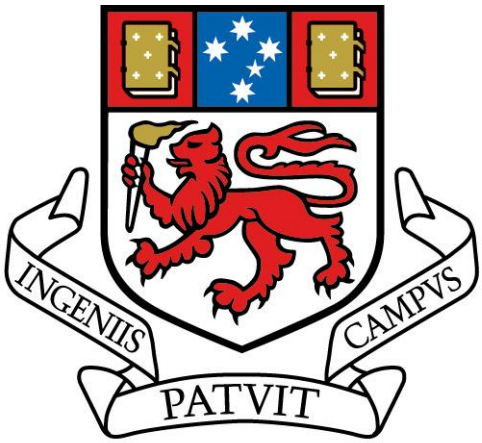

Natural Gestural Interaction for Geovisualisation

Simon Francis Stannus BComp (Hons)

Submitted in fulfilment of the requirements for the degree of Doctorate of Philosophy at the School of Engineering and ICT, University of Tasmania (November, 2015)

Declaration of Originality

This thesis contains no material which has been accepted for a degree or diploma by the University or any other institution, except by way of background information and duly acknowledged in the thesis, and to the best of my knowledge and belief no material previously published or written by another person except where due acknowledgement is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Signed:

Date:

Statement of Authority of Access and Regarding Published Work

The publishers of the papers significantly comprising Chapters 3 and 4 hold the copyright for that content, and access to the material should be sought from the respective journals. The remaining non-published content of the thesis may be made available for loan and limited copying and communication in accordance with the Copyright Act 1968.

Signed:

Date: 03/11/2015

Statement of Co-authorship

The following people and institutions contributed to the publication of work undertaken as part of this thesis:

Candidate	=	Simon Stannus,	University of Tasmania
Author 1	=	Daniel Rolf,	University of Tasmania
Author 2	=	Arko Lucieer,	University of Tasmania
Author 3	=	Winyu Chinthammit,	University of Tasmania
Author 4	=	Wai-Tat Fu,	University of Illinois at Urbana-Champaign

Paper 1

Chapter 3 was an extension of **Paper 1**:

> *Stannus, S., Rolf, D., Lucieer, A., & Chinthammit, W. (2011). Gestural navigation in Google Earth. In Proceedings of the 23rd Australian Computer-Human Interaction Conference - OzCHI '11. ACM Press, pp. 269-272.*

The **candidate** was responsible for designing and implementing the system and user experiment and prepared the initial manuscript draft. **Authors 1, 2 and 3** critically revised the manuscript.

Paper 2

Parts of **Chapter 4** were based on **Paper 2**:

> *Stannus, S., Lucieer, A., & Fu, W. (2014). Natural 7DoF Navigation & Interaction in 3D Geovisualisations. In Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology - VRST '14. ACM Press, pp. 229-230.*

The **candidate** was responsible for designing and implementing the system and user experiment and prepared the initial manuscript draft. **Author 2** provided data used in the experiment. **Authors 2 and 4** critically revised the manuscript.

Paper 3

Parts of **Chapter 4** were also based on **Paper 3**:

> *Stannus, S., Fu, W., & Lucieer, A. (2014). Natural 7DoF Input for 3D Navigation. In Proceedings of the 26th Australian Computer-Human Interaction Conference - OzCHI '14. ACM Press, pp. 216-219.*

The **candidate** was responsible for the core idea and initial manuscript draft. **Authors 2 and 4** critically revised the manuscript.

We the undersigned agree with the above stated "proportion of work undertaken" for each of the above published (or submitted) peer-reviewed manuscripts contributing to this thesis:

Signed: ___

Prof. Henry Duh	Prof. Andrew Chan
Supervisor	Head of School
School Of Engineering and ICT	School of Engineering and ICT
University of Tasmania	University of Tasmania

Date: 2nd Nov, 2015 — 2nd Nov 2015

Statement of Ethical Conduct

The research associated with this thesis abides by the international and Australian codes on human and animal experimentation, the guidelines by the Australian Government's Office of the Gene Technology Regulator and the rulings of the Safety, Ethics and Institutional Biosafety Committees of the University.

Signed: .....

Date: 03/11/2015

Abstract

As the power of computers has increased, the bottleneck for many tasks have shifted from the capability of computers to process information to the ability of humans to absorb this information and relay their input back into the computer accordingly — a cycle largely dependent on the interface through which users interact with the machine. The mouse and keyboard, which were designed for the computing tasks of decades ago, are still the dominant interface today, despite recent advances in natural interaction technologies and an ever-widening range of applications.

Geovisualisations are used for many applications, ranging from casual use by the general public to sophisticated research into complex phenomena. Being multi-dimensional and potentially complex, GIS datasets are the ideal candidates for applying new natural approaches to interaction. Gesture has long been associated with natural interaction, but existing research has failed to surpass the traditional desktop interaction paradigm in usage.

The aim of this work is to establish whether or not a gestural approach built on well-considered theory and refined by experimentation can provide a practical improvement to geovisual interaction. It also sets out to document any discoveries regarding requirements for both interaction design and implementation made in this endeavour.

The thesis begins with an introduction to geovisualisation outlining the types of data used and the growing areas to which it is applied, especially those of a 3D nature. Efficiency, learnability and comfort are identified as the key targets for improvement and it is argued that the ideal approach would be a bimanual interface following the principles of direct and simultaneously integral interaction.

An initial unencumbered prototype implemented using computer vision is presented along with the structure and results of a comparative user study in which this prototype was tested against existing devices for navigation in Google Earth. This pilot study uncovered outstanding technical roadblocks and yielded invaluable qualitative feedback.

Based on these insights, the case is made for *AeroSpace* — a technique built around a novel metaphor for spatial manipulation and navigation with the full seven degrees of

freedom. *AeroSpace* extends the two-point method of transformation commonly used with touchscreen devices to control navigation in immersive 3D spatial environments, using the 3D position and direction of the user's index fingers together with simple pinch and point hand poses. This approach was implemented using custom gloves in a test-bed built around NASA's World Wind virtual globe software. A description is given of a comparative user study involving tasks related to a high-resolution landscape model. Results from this study are presented that clearly show that users complete tasks that combine navigation with marking areas and data-points significantly faster when using *AeroSpace* compared to a popular commercial device specifically designed for 3D spatial interaction. The users also rated it as the significantly more natural and comfortable of the two approaches.

Altogether, this work presents evidence for a number of conclusions. It shows that the benefits of encumbered gestural interaction will continue to outweigh the disadvantages for the foreseeable future. It also presents qualitative and quantitative evidence to show that future gestural interaction schemes should follow the principles of direct bimanual and simultaneously integral interaction. However, it also demonstrates that users do not always expect natural metaphors and that physical navigation in particular is underutilised by new users.

This work also opens up new areas of potential future research. One priority is to identify ways of increasing the utilisation of virtual navigation and simultaneous 7DoF navigation. Also, the role of the level of directness in the success of *AeroSpace* remains not entirely evident. Immersive head-mounted displays would allow this and approaches to collaboration in 7DoF space to be suitably tested.

Acknowledgements

Thanks are due to the many people who have helped and supported me over the course of my doctorate.

I would like to start by thanking my supervisors, Dr. Ray Williams, Dr. Arko Lucieer, Dr. Winyu Chinthammit, Dr. Daniel Rolf and Prof. Henry Duh for the instrumental roles they played in the various stages of my doctorate.

I would also like to thank the administrative staff for their help, Julia Mollison for her help and advice on so many occasions and Raelene Nicholas for dealing with so many of my trivial questions regarding the TNE program.

I also owe much gratitude to Andrew Spilling, Christian McGee and Tony Gray for finding the time in their busy schedules to help with all things technical. Much is also owed to Bruce Andrews for teaching me the arcane workings of the VisionSpace, upon which my work relied so heavily.

I would also like to thank everyone else who brought sunshine to the windowless corridors of Building V, in particular Dean Steer for his infectiously jovial attitude, Matthew Springer for his irreverent quips and generous assistance looking over chapter drafts and Rob Rowe for his much-welcomed advice, stories and discourses on so many topics. I am also very grateful to all the students I have taught over the years for helping maintain my sense of purpose during some of the tougher periods.

I am also grateful to all those with whom I have had the pleasure of sharing an office, including Thomas Grayston, Jakub Dostal, Amanda Lunt, Zongyuan Zhao, Jahangir Kabir, Kane Lee and Liangjun Song, for all the conversations and welcome distractions from work. I would also like to thank Zhicong Lu, Mengxing Ao, Nawen De and Wenjia Nie for their friendship; their stay here felt altogether too short. I am also glad to have gotten to know Jaakko Hyry during his yet shorter stay and am very thankful for his warm hospitality in wintry Finland. I would also like to thank Callum Parker and Crystal Yoo for the fun that was had on our many outings.

I would also like to thank Robert Budzul, Tim Warren, Andrew Johnson, Sarah Wang and all the other members of the UTas Language Society for offering such a convivial source of respite from the long journey to my own academic *Fina Vinko*.

I also reserve special gratitude for my doctoral comrades, Mark Brown and Steve Neale, for sharing the ups and downs of the PhD experience and being such challenging opponents in table-tennis. I am also grateful to Mark for his tutelage in juggling and constant words of encouragement and Steve, the very model of a refined English gentleman, for being such an excellent wingman and host.

I would also like to thank my siblings — Gabrielle, Jean-Paul and Julien for their support and hospitality over the years, Madeleine for her generosity and assistance in all matters sartorial and especially Oliver for his immense help and advice during the thesis-writing process. Lastly, I am indebted to my parents for providing a roof over my head, home-cooked meals and unwavering support and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

So high, so low, so many things to know.

— Vernor Vinge, *A Deepness in the Sky*

1.1 Introduction

Geovisualisation is a domain worthy of interest for a number of reasons. Though it is has existed in some form or another for decades, it is only recently that graphics and network technology has advanced to the point that complex and realistic visualisations are widely accessible. The current trends in hardware development suggest that interaction techniques will be the focus of the next such shift.

1.2 Motivation

In particular, this shift will improve interaction and visualisation in 3D, which is inherently appropriate for the 3D environments that geovisualisations represent. However, convincing users to abandon long-established interaction paradigms and adapt to a new one presents a significant challenge. There also exists the danger that a single bad experience with one 3D interface will be enough to discourage a user from trying any others.

Therefore, as well as creating a more efficient workflow, it is important to design an interface that is easily-learnt and capable of appealing to new users. Techniques that are modelled after interaction that occurs naturally to humans more or less inherently satisfy these requirements, but have yet to establish their overall effectiveness for 3D interaction, though gesture in particular has shown its potential to enter the mainstream with the release of devices such as the Kinect and Leap Motion.

1.3 Aims

Thus, the central aim of this work is to design such a natural gestural interface for geovisualisation and validate its performance in quantitative terms with users who are largely unfamiliar with 3D interaction. To this end, it also aims to examine and test the underlying principles that determine the success of such interfaces.

1.4 Thesis Outline

The following is an outline of the structure of this thesis:

Chapter 2 gives an overview of the field of geovisualisation and the types of data and interactions it involves. It also touches on human-computer interaction theory, discussing in detail the factors relevant to this research before examining gesture as a

basis for interaction. It concludes by summarising the research questions this thesis sets out to answer.

Chapter 3 describes an initial attempt at a freehand gestural system for navigation in Google Earth using a stereo camera. It details the system's unique combination of image processing techniques and the results of user testing comparing it to existing interaction devices.

Chapter 4 logically argues from first principles the case for embodied 7DoF navigation and from this definition derives a metaphor and pose model that also better conforms to the design principles discussed in Chapter 2. It then presents a glove-based implementation and the results of its user testing in comparison with an existing commercial 3D mouse device.

Chapter 5 addresses the research questions by comparing the two gesture systems and their results to determine the degree to which implementation and fundamental interaction design factors affected their performance. It also discusses the implications of this research, mentions possible improvements and makes suggestion for future research.

Chapter 6 summaries the key conclusions of this work and their implications for future research.

Chapter 2 - Literature Review

How inappropriate to call this planet Earth when it is quite clearly Ocean.

— Arthur C. Clarke

2.1 Introduction

This chapter is intended as an introduction to the fields that this thesis is involved with. It is principally concerned with two such domains: Geographical Information Systems (GISs) and Human-Computer Interaction (HCI). Along with their intersection, different areas within those domains are discussed, all of which are spread across the following sections:

- Section 2.2 introduces geovisualisations, the types of data they involve and some of the applications for which they are used. It also describes the general interactions involved with such applications, why there is reason to believe they are open to improvement as well as the importance of such improvement.
- Section 2.3 outlines the areas in which this research endeavours to improve interaction with geovisualisations.
- Section 2.4 details fundamental HCI principles and how they relate to the interaction goals.
- Section 2.5 examines gesture and how it can be applied to this problem.
- Section 2.6 presents the fundamental questions that this research attempts to answer.

2.2 Geovisualisation

Geographical Information Systems (GISs) have existed for many years and have grown to cover a broad range of applications in a number of different forms. In some ways they are akin to any other information system that deals with heterogeneous data; however the characteristic that is common to all GISs is that any such non-spatial data point is associated with and seen as secondary to a spatial element. Indeed, this spatial element is powerful in that it can be used to strongly integrate otherwise unrelated data. (Maguire 1991)

According to MacEachren & Kraak (2001), maps have historically represented both the database and presentation of geographical information. However, with the advent of modern GIS systems, these two aspects have been separated and maps have become merely an interface through which the underlying data are accessed. Moreover, the dynamic capabilities of digital maps allow them to not only facilitate presentation, but

also information synthesis, analysis and exploration (Figure 1). The field that has emerged to deal with these possibilities is known as Geovisualisation.

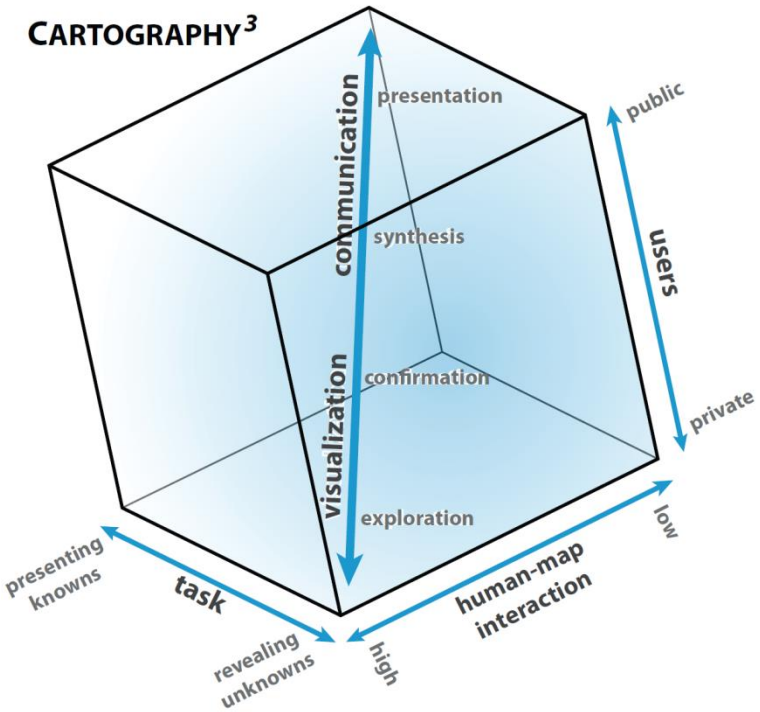

Figure 1: MacEachren and Taylor's (1994) multidimensional representation of cartography, showing visualisation as one direction through these dimensions (from reproduction by Roth (2013))

A study by Wisniewski, Pala, Lipford, & Wilson (2009) analysed the behaviour of a complex map visualisation and made the following recommendations:

- Reduce Complexity
- Encourage Interactivity
- Support Customisations

This research tackles the second point by attempting to improve the way in which interaction with geovisualisations is performed.

2.2.1 Geovisual Data Types

To understand the requirements of geovisualisations, it is important to understand the types of data they represent. These data are either raster or vector-based in nature.

One of the main sources of raster data in GIS is ***satellite imagery***. Such imagery may assist in digitisation or serve as the data itself, as in Google Maps, for example. In addition, some imagery is hyperspectral, representing a wider range of channels of light

than humans can see. ***Elevation models***, which model the elevation of a ground surface, are a related form of data. Though they usually occur as 2.5D height maps, which are fundamentally 2D images mapped to geographical space, they can differ from regular imagery in both their means of collection and visualisation. Laser scanning or 3D construction from imagery representing multiple points of view can be used to generate a digital elevation model (DEM).

Imagery and elevation models serve as a substrate onto which GIS users can add further abstract vector-based data, traditionally consisting of points, lines and polygons (Kersting & Döllner 2002). ***Points*** are one of the most fundamental geographical data types. While isolated points perhaps represent only a small part of traditional GIS datasets, they are becoming more common in user-generated geotags. Tweets alone produce millions of such location references per day (Hwang et al. 2013). ***Lines*** represents two or more points in series and can be used to describe either static features such as roads and tracks or the paths taken by objects as they move in time. ***Polygons*** represent the area contained by a looped series of points and are more commonly associated with traditional GIS, where they are used to represent everything from cadastres to ecological regions. Table 1 illustrates some of the uses of these data types.

While 2D maps are sufficient for many applications, there are a growing number of applications that map ***3D spaces***. Urbanisation in particular is a driving force behind many such applications (Ekberg 2007). Cities grow upwards as well as outwards, with some skyscrapers having over a hundred floors. Spaces are also carved out under the ground, with many subway stations resembling complex ant colonies. Furthermore, networks of tunnels and pedestrian walkways can allow people to walk for miles without touching the ground (Frampton et al. 2012). Figure 2 and Figure 3 illustrate the complexity of such situations.

		2D	3D
Vector Features	points	addresses landmarks individuals	birds marine animals aeroplanes submersibles
	lines	roads rivers	flight paths trajectories marine animal paths
	polygons / shapes	political areas cadastres	urban buildings subterranean networks
Raster Features		remote imagery land/vegetation classification	meteorological data geological data

Table 1: Example geographical features in 2D and 3D spaces

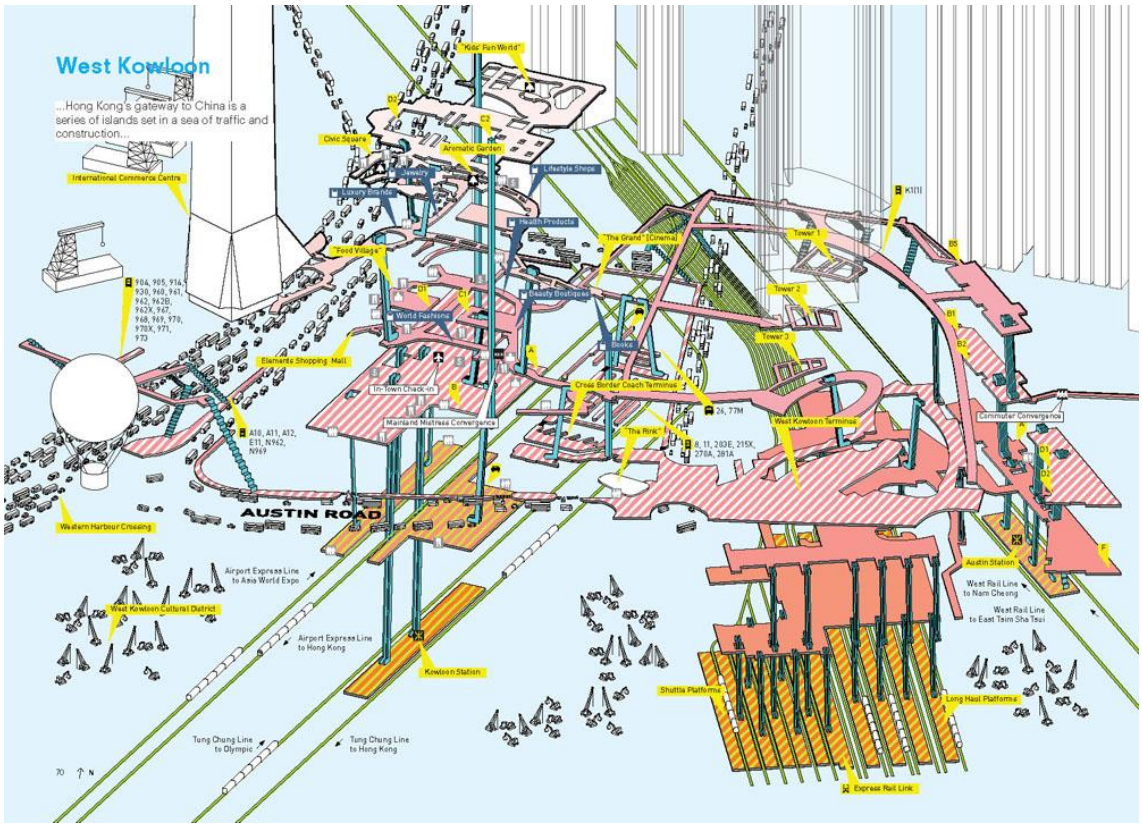

Figure 2: An image from Frampton et al. (2012) showing the complex network of tunnels and walkways in Kowloon, Hong Kong

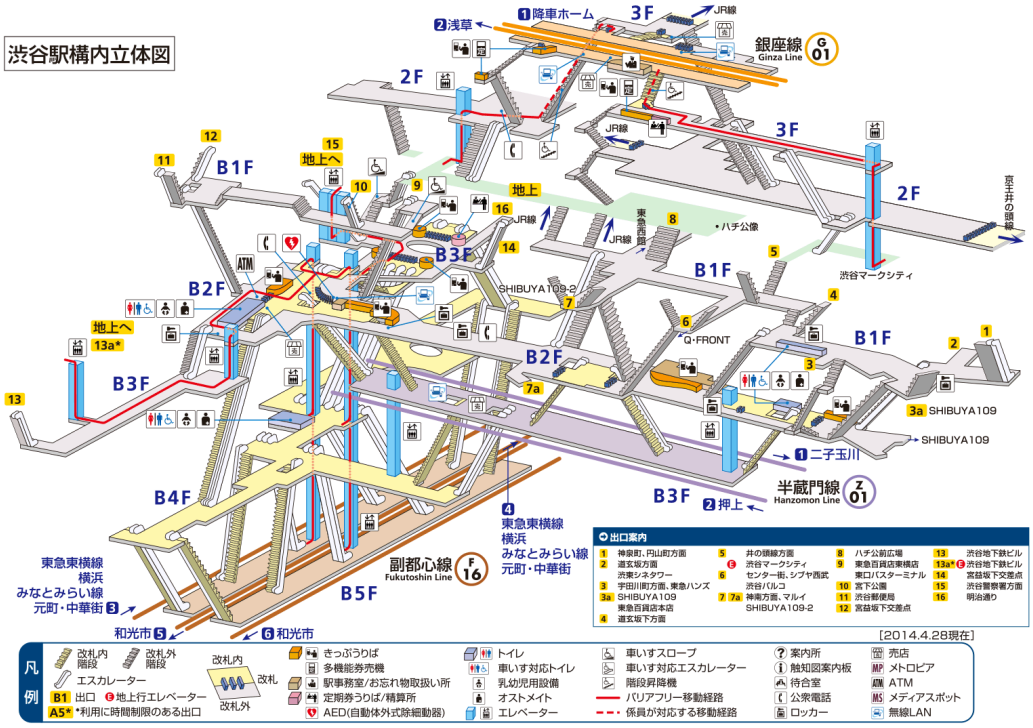

Figure 3: A map showing the complexity of the Shibuya station of the Tokyo Metro

2.2.2 3D Geovisualisation

Traditionally geographic data was visualised through maps, which presented 2D information in a static form. Computers and GIS software made it possible to have a dynamic viewpoint that could move between different latitudes, longitudes and scale. More recently improvements in hardware have prompted the development of virtual globe software that render a 3D model of the Earth's landscape and allow the viewpoint to be tilted and reoriented as well as moved. The most well-known of these is Google Earth, but there are a range of others, including NASA's World Wind and Microsoft's Bing Maps. While it may be unnecessary for some applications, Nielsen (2007) lists the following arguments for 3D geovisualisation:

- ***Simple transformations*** *between model, data and reality.*
- ***Presentation****: 3D models are natural and cognitively easier to interpret and are thereby suitable to communicate ideas and visions.*
- ***Exploration****: the human vision is made to quickly interpret a large amount of content or data in a scene. There are relations in a scene which the human brain percieve, often without being conscious of it. A large amount of data can therefore be communicated [...]*

- ***Immersion****: The user can be led through hardware interfaces to get the feeling of immersion into the scene and thereby to have a strong sense of being in a physical world. This has been used in adventure oriented models, as well as in product development of for instance engines.*
- ***Documentation****: Much geographical information contains height information that are only handled as additional information in 2D. In 3D, a more exact documentation can be performed.*
- ***Simulations and dynamics****: temporal simulations of 3D data can give new ways of studying complex processes in nature and society. In 3D they become more realistic simulation than in 2D.*

The remainder of this section focusses on 3D geovisualisation by describing some of the application areas that are dependent on it and the challenges that it introduces. While the work outlined in later chapters is chiefly concerned with the visualisation of naturally 3D geospatial data for a few specific applications, the broad range of applications discussed herein serve to underline the importance any improvement to interaction with 3D geovisualisations would have.

2.2.2.1 Planning

Planning is one such application (Jiang et al. 2003). Urban areas especially benefit from visualisation, allowing planners to rapidly evaluate the effects of planning decisions. Often, a two-dimensional representation is insufficient to deal with the factors that such planning must assess. One such factor is the effect new buildings have on the appearance of a city. For example, Nielsen (2007) cited examples where insufficient or erroneous planning visualisations led to unwanted changes to skylines that only became apparent after the respective multi-million dollar constructions were completed. In addition, 3D geovisualisation allows more abstract concepts, such as noise pollution (Kurakula 2007) to be more easily modelled and understood (Figure 4).

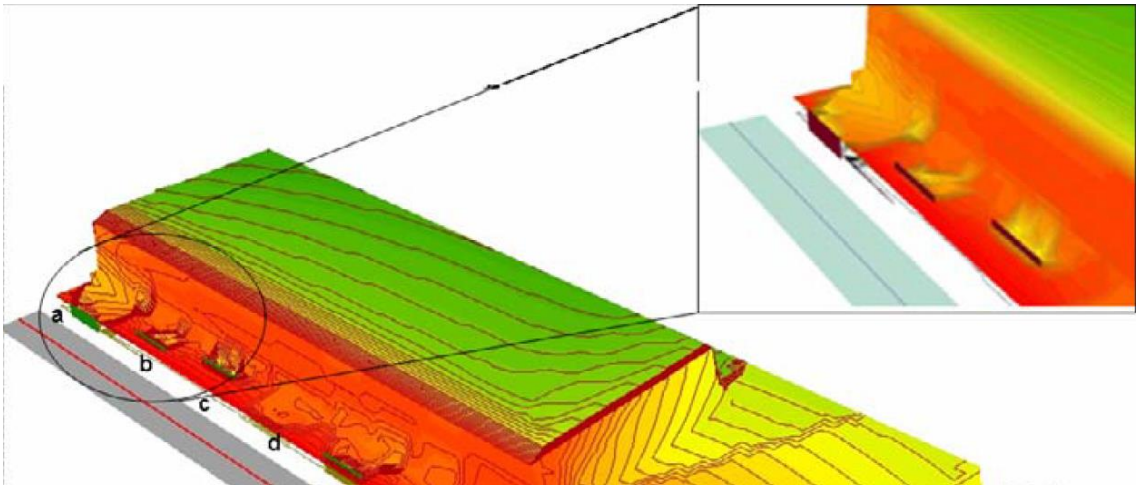

Figure 4: Image from Kurakula (2007) showing modelling of noise-reduction measures on a noise model of an urban environment.

2.2.2.2 Emergency Response

While urban planning deals with long timeframes of years and decades, 3D geovisualisation is also used in time-critical applications like emergency response, where every second counts.

Lee and Zlatanova (2008) argued in support of the necessity of representing the 3D aspect of geospatial data in emergency response applications where rescues or evacuations may need to be carried out in complex urban environments. In such situations, the speed with which the interaction interface allows decisions to be made is crucially important; bottlenecks could potentially delay response times and thus cost lives.

Kwan and Lee (2005) showed that for emergency response teams such delays faced within multi-storey buildings can be greater than those faced on the ground in reaching the building. They also showed that 3D GIS can be used to considerably reduce these times. Lee (2007) went further by claiming that the standard data models for 3D spatial analysis were insufficient for 3D visualisation of navigation in an emergency and required a Navigable Data Model.

2.2.2.3 Security

Closely related to emergency response is the area of security, which can also be concerned with a 3D representation of the world. For example, VanHorn & Mosurinjohn (2010) showed its potential for visualising the viewsheds (i.e., the volumes representing possible lines of sight) of important locations to analyse the threat posed by

sniper attacks. They argued that given the existence of overhanging obstructions, 2.5D modelling was insufficient for calculating the viewsheds, let alone visualising them.

2.2.2.4 Mining

Another area in which 3D geovisualisation is becoming increasingly important is geological modelling. Because mineral deposits are spread through the depth of the ground as well as across the two standard cartographical dimensions, standard maps are not sufficient for their visualisation.

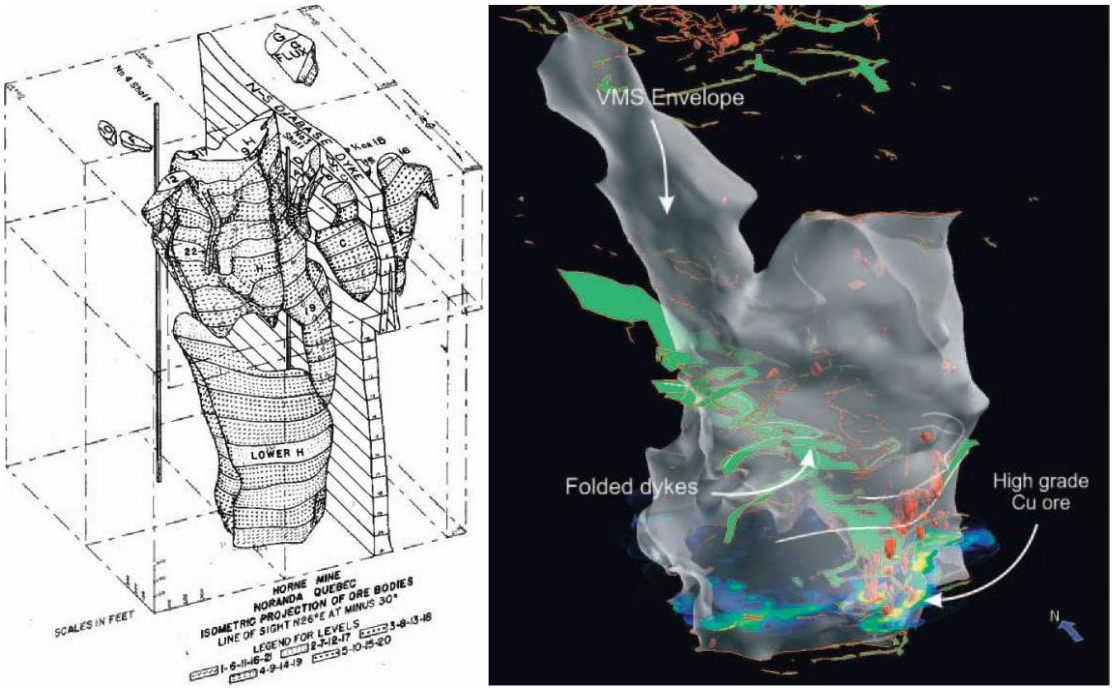

Figure 5: An early visual representation of an ore body (Price 1934) and a modern digital equivalent of the same body, from (de Kemp et al. 2011)

In recent years, work such as that by de Kemp et al. (2011) and Wang et al. (2011) has shown the potential of 3D geovisualisation to assist in identifying the spatial relationships between geological bodies (Figure 5). Similarly, Shao et al. (2011) identified 3D visualisation of geological bodies as important to the future efficiency of oil and gas exploration.

2.2.2.5 Challenges

Slocum et al. (2001) recognised the capacity of 3D geospatial virtual environments (GeoVEs) to provide immersive interaction with the "*look and feel*" of the real world, but listed a number of research challenges:

- Determine the situations in which (and how) immersive technologies can assist users in understanding geospatial environments.
- Develop methods to assist users in navigating and maintaining orientation in GeoVEs.
- Develop suitable methods for interacting with objects in the GeoVE.
- Determine ways in which intelligent agents can assist users in understanding GeoVEs
- Determine ways in which we can mix realism and abstraction in representations to influence cognitive processes involved in knowledge construction.
- Developing support for interpreting and understanding spatial trends and patterns in GeoVEs.

Since then, there has been progress in a number of these areas. For example, there have been a number of examples of research on the application of display technologies to increase immersion (Knust & Buchroithner 2014).

Others have also worked on representing abstraction and realism. The best example of this is the combining of a non-spatial dimension with the two standard cartographical dimensions to produce more 3D visualisations from data that is not inherently three-dimensional. This might take the form of a fake elevation model representation for raster data or 3D lines "*through the air*" for 2D line data that is supplemented with a third coordinate such as time (Kwan & Lee 2003). For example, Wolff & Asche (2010) demonstrated the usefulness of this approach for analysing crime scene distribution.

However, there is also evidence that the current standards of interaction require work. Shepherd and Bleasdale-Shepherd (2008) explored the interaction capabilities of video games and assessed their potential in virtual geographical environments. They defined VGEs as GIS and data visualisation software and stated that due to the increasingly 3D nature of visualisations "*a radical rethink of VGE user interfaces will be necessary*". They mentioned input based on "*body gestures and motion*" as one direction for such progress, with "*more cost-friendly versions of the body-tracking technologies*" a requirement.

Zhou & Guo (2011) created a stereoscopic 3D virtual reality simulation of the terrain, shafts, tunnels and ore bodies of a mining site. While they recognised the increased

level of immersion their simulation allowed, one of their main conclusions was that the mouse and keyboard were insufficient for interaction and further research was required to enhance it.

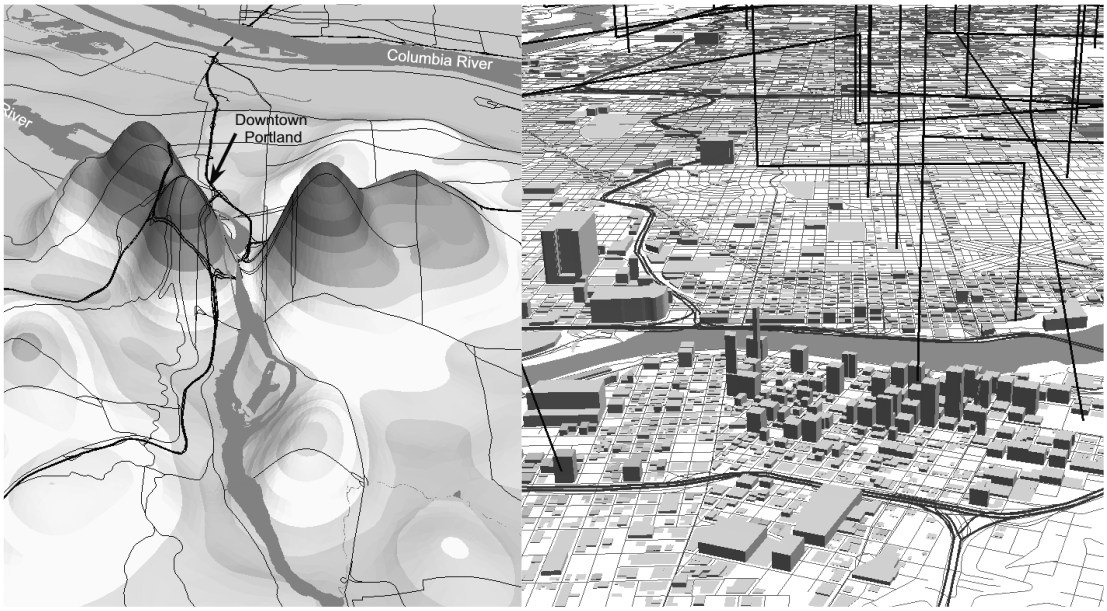

Figure 6: Examples from Kwan et al. (2003) of a non-spatial aspect of data being mapped to the Z axis, either as a variable dependent on the other two dimensions, such as with a height map (left image), or as an extra ordinate in vector data, such as time in plots of movement (right image)

Figure 7: An example from Wolff & Asche (2010) of the Z axis being used to represent the temporal aspect of geographical data

2.2.3 Interactions

Before specific improvements can be proposed, it is necessary to outline the kinds of interaction involved with geovisualisations. Vatavu et al. (2005) described interaction

with virtual environments as falling into three distinct categories: selection, manipulation and navigation.

Navigation, the moving through the virtual space, is a fundamental aspect of visualisation. It can be seen as a kind of positioning task, since it involves, in essence, moving a virtual camera position through 3D space. However, much data is both spatial and temporal in nature - that is it is spread across time as well as space. For example, the position data of an animal that is being tracked by a GPS tag could be updated regularly over many months. Similarly, climate data varies at scales from decades to millennia. Because of this, navigation needs to incorporate movement through not only space but also time. Swan & Gabbard (2003) have suggested that different navigation paradigms are required for the different applications of spatial navigation, from navigating GeoVEs at low and high-scales to moving through abstract scientific data.

Selection is the process of selecting particular objects or features to provide context for further tasks. The data being selected could be any of a range of different data types.

Manipulation is the changing of the properties of selected objects and may take a number of forms. One of the most fundamental in general spatial interaction is the moving of the object or its constituent parts; for example, moving a point, redefining a boundary or perhaps even remodelling a virtual landscape.

GIS in general and more specifically geovisualisations focus heavily on the visualisation aspect that covers more interaction than just spatial navigation. For example, Crampton (2002) list querying/data-mining, brushing and filtering/highlighting as the main types of interaction. These involve some form of selection of parameters, areas, operations and attributes and the results being displayed to the user.

2.3 Goals

In order to assess the potential of new interaction techniques and make comparisons with existing ones, it is necessary to choose which goals to aim for; i.e., which metrics to use for evaluating interaction. This section briefly describes the metrics that are considered relevant for this work.

2.3.1 Efficiency

Time is perhaps the most obvious metric for the success of a user interface; the less time it takes a user to complete a task, the more they can achieve in a given timeframe. Ideally, a GUI should be designed so that the time taken to switch between modes and effect actions is optimised to be as small as possible. As the work of Fitts (1954) showed, there is a limit to how fast a human's motor system can accurately control movement to a certain position. Fitts' experiments showed that the time taken for a person to move a stylus or similar object using his/her arm to a one-dimensional target is logarithmically related to the ratio of the distance to the target position to the size of the target. This phenomenon is known as Fitts' Law and allows the capacity of a human motor system to be defined in terms of bits per second (bps), which Fitts measured to be around 10bps for human arms. Fitts' Law has since been expanded to two-dimensional mouse pointing tasks (Accot & Zhai 2003) and is widely used as a basis of evaluating the performance of interfaces, such as in the works of May (2004), Elmqvist & Fekete (2008) and Sweetser et al. (2008). This goes to show the importance of minimising the distance and accuracy requirements of any mouse input to graphical user interfaces (GUIs). However, in standard GUIs there is a fundamental trade-off between screen real-estate and how easily buttons can be clicked, so there are limits to how far this can be optimised, especially in the case of more complex interfaces where there might be hundreds of different commands.

2.3.2 Learnability

Another important factor in interfaces (at least for short term users) is their learning curve. Nielsen et al. (2004) state that gestures should be intuitive and easy to perform and remember. However, Wigdor & Wixon (2011) describe new interfaces as presenting a *gulf of competence* to users who are already invested in an existing interface for the same task. They argue that even when they are aware that a particular interface with which they are not familiar may be more efficient in the long term, users are reluctant to invest the time in making the switch.

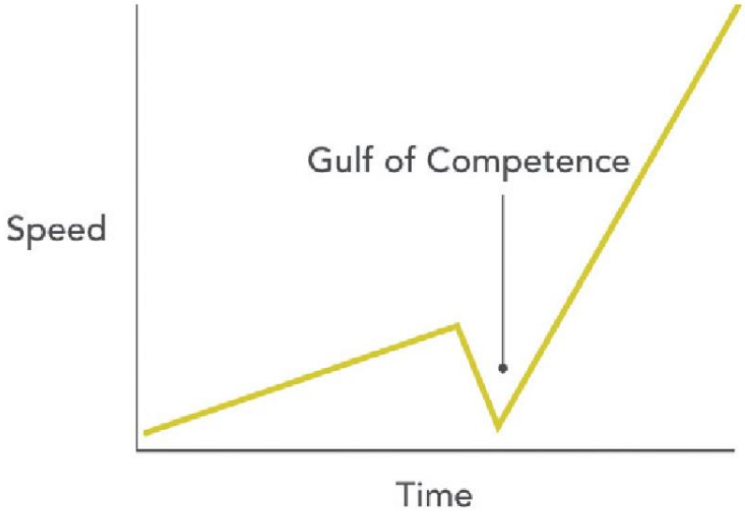

Figure 8: Wigdor & Wixon's (2011) Gulf of Competence

Ideally, an interface should be highly intuitive, so that the learning period is reduced, allowing it to quickly pay for itself in terms of efficiency gains.

2.3.3 Comfort

Comfort is also important, since even if an interaction method is efficient and easy to learn, it still might be impractical for prolonged use due to strain on the user's body. This can be difficult to validate in prototype designs, since certain injuries may only be apparent after long term use, though many interaction techniques quickly result in strain that is noticeable to the user. Nielsen et al. (2003) show that gestures designed without taking human biomechanics into account can cause too much physical stress to be useful.

2.3.4 Conclusion

Of the aforementioned metrics, efficiency is perhaps the easiest to measure, since users can be given an objective task and the time taken to complete it can be used as a measure. However, comfort is very much a subjective quality, so it is best determined through user feedback. Learnability can be ensured by making sure that users have not been previously exposed to the system being tested and that the duration of testing is kept to a minimum. This way, any system with a long learning curve will fail to produce good results.

2.4 Human-Computer Interaction Principles

As well-defined as these metrics are, it is not necessarily immediately clear how to design an interaction technique that will maximise results according to these measures.

To assist in understanding the potential of any theoretical technique, this section looks at the important interaction principles that have the potential to explain connections between the design and the target goals.

2.4.1 Naturalness

One of the ways to achieve these goals is to design the means of interaction to be as natural as possible. A natural interface is one that imitates the way people interact with the real world. One of the main advantages of such an approach is that it minimises the amount of learning required to become proficient with it.

2.4.2 Directness

While not necessarily a factor in its own right, the degree of *directness* (Shneiderman 1983) with which interaction is mediated affects the other variables. With a few exceptions (such as controlling the acceleration and steering of a car), our interaction with the real world around us can be considered direct; that is, our effect on the environment is directly proportional to our movement or that of our limbs. In this sense, directness can be considered synonymous with naturalness, and thus intuitiveness. Despite this, indirect methods, such as rate-controlled methods like pointing in the direction in which to move (Schlattmann et al. 2009), are widely used for interaction with virtual environments. Some claim that they are less intuitive to use, citing the difficult mental transformation required to control them (Zhai 1995).

A common distinction to make is that between *position-controlled* and *rate-controlled* techniques. Many studies (e.g., the work of Kulik, et al. (2009) and Oakley & O'Modhrain (2005)) show the relative efficiency and accuracy of the former, but truly direct interaction is just a subset of position-controlled techniques.

For spatial interaction, there is generally an area or volume of space in which the position of the hand or interaction device is mapped in some way to the virtual space in which the user is working. Song and Takatsuka (2005) called these two areas the *interaction* and *information* spaces. Interaction is direct if these spaces are equivalent; i.e, the user's hands (or other effectors) occupy the same physical space as the particular point they have an effect on.

May (2004) describes the link between movement of a pencil and the resulting writing on the paper as an example of directly-mediated interaction. On the other hand, Song

and Takatsuka describe the use of mice as an "*indirect mapping*", in which the user has to map their intention into the mouse movement that maps to the corresponding result.

In properly direct interaction, as with the user's conceptual model of manipulation (Kulik 2009), there is a kinaesthetic correspondence between the motion of their hand and the resulting impact on objects. This allows the user to take advantage of their body's natural feedback of the position and orientation of its limbs (Mine et al. 1997). This feedback, known as *proprioception*, is what makes it possible for users to perform some interactions blindfolded, even in the absence of touch. Illustrating its importance, real medical cases show that without proprioception, basic activities such as walking are impossible without years of training and even then rely on visual feedback as a fallback mechanism, meaning constant eye contact with the limbs in question is required (Robles De La Torre 2006). Proprioception is complemented by further feedback from the vestibular system, which is based in the inner ear and provides information on the movement and orientation of the head to the brain. Used together with visual stimuli, this allows us to map proprioception's body-relative data to our spatial model of the real world.

According to Mine et al. (1997), interfaces that allow users to make use of proprioception allow for better control and precision of manipulation and are the next best thing to those that give full haptic feedback. Indeed, there is plenty of experimental evidence to show that indirect interaction can negatively affect realism (Moehring & Froehlich 2011), performance times (Knoedel & Hachet 2011) and overall user experience (Özacar et al. 2013).

In order to evaluate input techniques in terms of their directness, it becomes important to define it precisely. Zhai and Milgram (1998) related directness to the mathematical simplicity of the transformation between *control space* and *display space*.

If c is the control space and d is the display space, then this mapping m can be represented as a function: $d = m(c)$. Table 2 shows a rough ordering of different mapping types and simplified versions of their defining mapping function, from most to least direct.

Mapping		Function	Terms
Position	Direct	c	Control Space c
	Aligned Offset	$c + t$	Translation Vector t
	Invariant Offset		
	Linear (CD Gain)	$s.c$	Scaling Scalar s
	Rotated	Rc	Rotation Matrix R
Rate		$\int m(c)\, dt$	

Table 2: A taxonomy of levels of directness, from most to least direct

Fully-direct interaction represents how humans fundamentally interact with objects in the real world. Indirect interaction by means of an offset occurs when we use certain tools (e.g., hitting a nail with a hammer or writing with a pencil). Generally these offsets are *aligned* with our hands or arms, in that if we rotate our hands or arms, the direction of the offset rotates with them. However, there are real-world examples, such as marionette puppets, where such an offset is invariant to rotation. The distinction between these two levels of indirection shows the limitations of modelling grasping hands as just a positional problem with three degrees of freedom.

Interaction where the scale between hand and effect is not 1:1 is perhaps less natural, but can occur when levers scale (and offset) interaction from a closer to a further point on a rigid object. Tongs are an example of a utensil where such scaling is used to magnify the extent of our grasp, whereas the movement of the load in a wheelbarrow is a minification of that of the user's hands. There are perhaps more examples where the pivot occurs between the input and output of force, causing the scale to be negative (e.g., pliers). A less commonplace but more striking example of scaled interaction in the physical world is that of pantographs (Figure 9), which are used to duplicate designs at different scales as they are drawn or sculpted.

Figure 9: A pantograph used for scaling down etchings

A good example of a fully-direct input mapping would be that of touch-screen devices; when a user touches their tablet's screen, they expect their interaction (be it clicking a button, selecting an object or outlining a shape) to occur where their touch has occurred.

In reality, most mappings are a combination of these elements. For example, the coordinate system of a mouse is offset and rotated from the horizontal surface of a desk to the upright plane defined by the monitor. Some amount of (often non-linear velocity-based) scaling is also involved, since there is no direct constraint between real world distance and screen pixels.

2.4.3 Absolute and Relative Interaction

A further distinction can also be made between absolute and relative mappings. While these terms are usually used to describe mapping a cursor's position, the same principles can be applied to navigation. With absolute interfaces, any position and orientation in the interaction area will always map to the same corresponding virtual coordinates. On the other hand, relative interfaces update the current virtual coordinates according to the relative change from the last point, possibly with a non-linear relationship between interaction and virtual velocities. While relative interfaces are susceptible to inaccuracies being compounded and may not seem as natural to users as absolute ones,

they might be necessary when raw input data is relative (such as IMU sensors or mice) or where absolute data is insufficiently accurate (Sweetser et al. 2008).

In either case, the problem arises of how to deal with large virtual spaces, where the scale of navigation is either too small, causing the navigation to be limited by the bounds of the interaction space, or too large, making it impossible to accurately effect and/or detect a motion small enough for some intended interaction. Relative approaches can get around this by allowing the user to temporarily reposition in the interaction space without updating the information space. The cycle of lifting and repositioning required to operate a standard computer mouse is a good example of this. However, this may be tedious and exhausting if the user has to continually reposition their hand in the empty area in front them. Also, there is the question of how to signify when and when not to track the hand's motion. O'Hagan et al. (2002) suggested use of a grasping pose, noting that it closely matches the way in which people interact with real objects. Similarly, they suggested a relative grasp and release method for rotation.

2.4.4 Simultaneous Degrees of Freedom

Spatial interaction is an inherently multidimensional task. Even the most basic scenario of a flat map on a desktop computer involves two dimensions of data and three to control the user's viewpoint (x, y and zoom). With any such multidimensional interaction, the dimensionality of the device and technique is important. This dimensionality can be considered the number of degrees of freedom that can be controlled *simultaneously*. If the input device's degrees of freedom are either fewer than those of the task or they cannot be performed simultaneously (e.g., the scroll wheel of a mouse that is too awkward to use during normal dragging), the motion must be broken up into multiple stages, costing significant extra time. While having too few can limit interaction (e.g. Darken and Durost (2005)), some, such as Bowman (2013), argue that having too many can require unnecessary effort from the user. Moreover, Jacob et al. (1994) extended the notions of integral and separable dimensions (Garner & Felfoldy 1970) to cover spatial interaction and argued that optimal efficiency occurs when the integrality of input dimensions correctly matches that of the task, meaning that having the correct number of degrees of freedom alone does not guarantee efficiency.

2.4.5 Bimanuality

Since most interaction is performed using hands, one way of increasing the simultaneous degrees of freedom is to design a bimanual technique, i.e. one that uses both hands. Owen et al. (2005) argued that bimanual manipulation can lead to improved task performance, firstly because time can be saved by parallelising tasks and reducing the need to switch between modes that might require interaction in spatially separated areas, and secondly because of cognitive benefits due to the higher bandwidth that two hands make available for epistemic actions, which uncover information from different possible interactions that would otherwise be hard to compute mentally. Indeed, there is evidence that shows bimanual techniques can produce faster performance (Balakrishnan & Kurtenbach 1999), less errors (Kulik et al. 2009), more proprioceptive cues (Capobianco et al. 2009) and improved comfort (Bi et al. 2012).

2.5 Gesture

Having analysed the various factors that are important to the effectiveness of spatial interaction, the next step is to use that information to guide the design of an interaction technique. Gesture is an interesting domain of interaction that is increasingly the focus of research and commercial development. One major advantage of gestural interaction is its potential for a reduced learning curve; by mimicking our direct interaction with the real world, gesture interfaces can remove the need for users to spend time learning new interaction metaphors. In addition, such natural gesture techniques have already been shown to outperform traditional techniques in some areas (Rizzo et al. 2005).

For example, LaViola (1999) noted the suitability for gestures in navigation of visualisations in Virtual Environments (VEs) and according to Vatavu et al. (2005), the hand gesture approach is the ideal interaction technique for selection in virtual environments because it can be "*implemented in a way that closely mimics real-world interactions*". Similarly, O'Hagan et al. (2002) describe hand gestures as an "*intuitive*" way of reducing the user's cognitive load, so long as the right set of gestures is chosen.

Such observations make gesture a worthwhile starting point for further investigation. This section briefly explains what gesture is and the challenges and techniques for its implementation.

2.5.1 Classification

Before the techniques for recognising and interpreting gestures are considered, it is important to first examine the different forms of gesture interaction and determine which are the most appropriate for this application.

According to Mitra and Achayra (2007), gestures can have many meanings dependent on their position, path, produced symbol and emotional quality. Vafaei (2013) went further by summarising previous classifications to produce a taxonomy of gesture made up of 11 dimensions arranged in two groups:

Gesture Mapping	
Dimension	**Classes**
Nature	Manipulative | Pantomimic | Symbolic | Pointing | Abstract
Form	Static | Dynamic
Binding	Object-Centric | World-Dependent | World-Independent
Temporal	Continuous | Discrete
Context	In-Context | No Context
Physical Characteristics:	
Dimension	**Classes**
Dimensionality	One | Two | Three | …
Complexity	Simple | Compound
Body Part	Hand | Arm | Head | Foot | …
Handedness	Dominant | Non-Dominant | Bimanual
Hand Shape	Open | Pinch | Thumbs-Up | Fist | …
Range of Motion	Small | Large

Table 3: The gesture taxonomy of Vafaei (2013)

This illustrates that gesture is very much a heterogeneous class of interactions and that care must be taken to choose the correct mapping for any particular task. Of particular importance is the *Temporal* aspect. It divides gestures into two classes: *Continuous* and *Discrete*. Discrete gestures are instantaneous actions that can be thought of as the gestural equivalent of GUI buttons or keyboard commands. Continuous gestures are performed over a length of time and allow for gradual actions. They dovetail well with gestures whose *Nature* is *Manipulative*, where the movement of the hand directly maps

to the movement of an object or entity; i.e., direct interaction. As already highlighted in section 2.4.4, the dimensionality will likely need to have at least three axes to suit the 3D nature of geovisualisation.

While a continuous manipulative gesture may address the spatial element of geovisual interaction, there remains a need for discrete action, for which the remaining dimensions and classes will need to be considered.

2.5.2 Recognition

There are a number of different existing techniques for hand gesture recognition. Each has its own advantages and disadvantages according to different criteria. The purpose of this section is not to settle on one particular technique but rather to clarify the limitations of each so that the rationale behind their use in later chapters is understood.

2.5.2.1 Gloves

One popular technique is the use of glove devices for input. Glove-based technology has the advantage of being highly accurate and does not suffer from most of the problems of vision-based approaches. However, data gloves can cost thousands of dollars, take considerable time to set up and restrict the user's movement. (LaViola Jr. 1999)

2.5.2.2 EMG

Another approach is to use surface electromyography (SEMG) to measure the electrical signal sent from the brain to the muscles that control the movement of the user's fingers. Shrirao et al. (2009) showed that filtering the SEMG signals and using them as inputs to a committee of neural networks (CNN) led to a reasonably accurate system. They noted that such a system would be less of an encumbrance than traditional glove systems. Another interesting result they reported was that data from the EMG sensors reflected movement in the user's hand approximately 0.2 seconds before it occurred. Xu et al. (2009) showed that EMG can be useful in supporting accelerometer data by segmenting the start and end of gesture. Their system for interaction with a virtual Rubik's Cube using such a method detected 18 different gestures with 91.7% accuracy.

A few years ago, research labs were the only ones to apply EMG technology to human-computer technology, but recent devices such as the Myo have made it affordable and practical for consumer use.

2.5.2.3 Handheld Devices

Devices that can be held in the hand are one way of obtaining hand position and orientation information. Sweetser et al. (2008) developed a handheld device that contains a camera, which detects the position and signal strength of infrared markers placed next to the screen. The device uses this information to calculate the 6 degrees of freedom that represent the position and orientation of the device. While hand-held devices may be an appropriate solution to tracking position and orientation, they tie up the fingers of the user's hand, preventing them from performing gestures. Instead, buttons can be used for discrete actions, though arguably at the cost of naturalness. In recent years, one of the most widely used examples of handheld device is Nintendo's Wii-mote.

2.5.2.4 Computer Vision

Computer vision is a popular way of detecting and comprehending gestures. The main advantage of such an approach is that it does not require expensive and sometimes inconvenient equipment (Hassanpour et al. 2008). May (2004) states that computer vision is a relatively flexible means of gesture recognition and that video frames contain a large amount of relevant information. He also notes the potential for computer vision to be passive and ubiquitous. Indeed, all the other approaches so far examined require some kind of physical constraint to the user's hand and some rely on devices that are not widely available. Because computer vision obviates these requirements, it is the method of choice for this research.

LaViola (1999) recognises four important components in a vision-based hand gesture recognition system: placement and number of cameras; visibility of the hands; feature extraction, and; classification of gestures from those features.

The first component is essential to overcoming the problem of occlusion, where the important features of a user's hand are blocked by other objects or even the hand itself. If multiple cameras are used, depth information can be gained and visibility of the hand can be increased.

2.5.2.4.1 Visibility

The second component relates to the ability of the system to accurately recognise the pixels that represent the hands in the video. This process is known as segmentation. A simple colour or intensity-based thresholding of the video frames according to the

colour of the user's skin is not likely to separate the hands from the background successfully in most situations due to irregularities caused by the different effects of lighting. Requiring the user to wear gloves that are brightly coloured (Keskin et al. 2003) or have LEDs is one way of making segmentation easier. Making the background a uniform colour can also be useful (LaViola Jr. 1999). However, these techniques improve accuracy at the cost of extra constraints to the user's interaction; it will not always be feasible for users to work in front of a fixed-colour backdrop or wear special gloves.

Wachs et al. (2006) developed a system for hands-free navigation of medical information. It was designed to be used in a sterile environment, where use of a keyboard is not possible. Their system used the CAMSHIFT algorithm, a histogram-based algorithm, to segment the pixels of the hand. To choose the value ranges for CAMSHIFT, the user had to calibrate the system by placing their hand in a specified position and then moving it around. The movement of the hand over the static background is detected, allowing the colour values of the hand's pixels to be defined. The motion is detected by comparing the colour values of pixels at the same location in different frames. Because the background is mostly static, the area with major changes in pixels values can be assumed to be where the user's hand is. This approach can be extended to estimate the optical flow of an area by searching for the nearby area where the corresponding pixels are least different, giving information on how far and in which direction objects and features have moved.

2.5.2.4.2 Feature Extraction

Feature extraction covers the range of methods for extracting measurable information, or features, from raw data. Hassanpour et al. (2008) divide features into three levels: high-level features, the image itself and low level features measured from within the image. According to this definition, high level features are obtained by searching for the best set of parameters of a 3D model of the hand to fit the data obtained from camera(s). This fitting process can match a particular pose of the model against the edges found in the frame from the camera or, in the case where multiple cameras are use, the region of hand voxels (volumetric picture elements) constructed from the silhouette images from multiple views.

Bowyer et al. (2006) mention a way of detecting depth for the purposes of 3D face recognition; if a pattern of light is projected on to the object and a camera detects the deformed pattern, a 3D model can be reconstructed. However, this approach is not suitable for this research as it requires somewhat constrained ambient illumination and the projected light necessary might be distracting or harmful to the user.

Lee et al. (2008) were able to implement pointing detection in a simple manner. The hand was defined to be the blob with the contour after thresholding according to a colour histogram. The fingertips were then found by examining the convexity of the blob's contour. The furthest of those fingertip features from the centre was deemed the tip of the pointing finger, and the line extrapolated from the palm and this fingertip was used as the pointing direction. The 3D positions of those two points were obtained using the disparity map from a stereo-camera system.

Shin et al. (2003) use a measure of the hand segments area as a simple way of determining whether it is open or closed into a fist. This information was used to control when gestures begin and end.

For the purposes of this research, the most important features will likely be the relative positions of the user's fingertips. A technique similar to that of Lee et al. (2008) would seem the most appropriate way of achieving this.

2.5.2.4.3 Classification

The fourth of LaViola's components is the way in which the extracted information is used to classify gestures. This component may not be so relevant to detection of the hand's position and orientation, but is important for the recognition of poses.

Possibly the most popular classification technique is Artificial Neural Networks (ANNs). May (2004) used an ANN which took the relative positions of the user's fingers as inputs. His system was able to classify 8 different poses reliably, with the exception of 2 similar poses being confused for each other.

Malerczyk and Engleke (2009) used a Naïve Bayes classifier for classifying hand poses, citing a balance between the time taken to build up a model of the user's hand and the time taken to classify gestures. They were able to achieve approximately 95% accuracy in classifying 3 different poses.

O'Hagan et al. (2002) used a statistical method based on logistic regression for classification, using the location of fingertips as well as the valleys between fingers as features. The accuracy varied between the poses from 73% to 100%.

One popular method for recognising gestures defined by the paths they make is Hidden Markov Models (HMMs). HMMs model the likelihood of states in a given process, when the probabilities of transitions between states and the probabilities of states causing observations are known. For gesture recognition, the motion of the hand is usually discretised into a series of translations in one of a finite number of directions, which represent states in the HMM. Keskin et al. (2003) and Elmezain et al. (2009) all use this approach, achieving over 94% accuracy classification of gestures.

These different classification methods will be further investigated to find which is the most appropriate for this research. The key requirements are that it be robust, accurate and able to perform in real-time.

2.6 Research Questions

From this review it is clear that there remain gaps in the body of knowledge relating to the application of natural interaction to geovisualisations. Chiefly, the following question needs to be addressed:

Can a gestural approach based on natural interaction principles improve interaction with geovisualisations?

Furthermore, this leads to the following questions:

- To what degree are each of the principals of natural interaction critical to the success of such interaction?
- Which technologies and techniques can be used to enable such interaction and what issues arise in its implementation?

The remainder of this thesis attempts to expand the knowledge in this area by tackling these questions.

2.6.1 Research Generalisability

While in some ways these questions regarding geovisualisation are generalisable to 3D interaction in general, it cannot be assumed that the specific constraints of

geographically distributed data do not significantly alter navigation and interaction. Though geovisualisations may include abstract 3D data, by definition they remain grounded to the 2.5D context of the virtual globe. This suggests that, unlike abstract 3D environments, navigation in geovisualisations follows certain patterns, closely related to the globe's surface in terms of both position and orientation. For 2.5D datasets, the largely convex nature of the earth means that spatial data are much less likely to obscure each other, especially at large scales.

Also, the fact that humans have evolved and learned to have a natural understanding of geographical features, if only at a relatively local level, means that geovisualisations may also have inherently less issues with cognitive load than other cases of 3D interaction.

Chapter 3 - 3D Position-Controlled Navigation with Computer Vision

Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun. Orbiting this at a distance of roughly ninety-two million miles is an utterly insignificant little blue green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea.

— Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

Parts of chapter 3 have been removed for copyright or proprietary reasons.

Chapter 3 is an extension of the following paper: Stannus, S., Rolf, D., Lucieer, A., Chinthammit, W. (2011). Gestural navigation in Google Earth. In Proceedings of the 23rd Australian Computer-Human Interaction Conference - OzCHI '11. ACM Press, pp. 269-272. 978-1-4503-1090-1

Chapter 3 - 3D Position-Controlled Navigation with Computer Vision

Far out in the uncharted backwaters of the unfashionable end of the western spiral arm of the Galaxy lies a small unregarded yellow sun. Orbiting this at a distance of roughly ninety-two million miles is an utterly insignificant little blue green planet whose ape-descended life forms are so amazingly primitive that they still think digital watches are a pretty neat idea.

— Douglas Adams, *The Hitchhiker's Guide to the Galaxy*

3.1 Introduction

This chapter represents the first attempt of this research to implement and evaluate improvements interactions with geovisualisations. As such, it does not attempt to tackle a full range of use cases, instead focussing on applying 3D position-controlled natural interaction to the most central type of interaction — navigation — with unencumbered freehand (i.e., gestural) input put forward for examination as the ideal way of achieving this. The structure of the chapter is as follows:

- Section 3.2 outlines the limitations of the current approaches to navigation.
- Section 3.3 discusses previous attempts to apply gesture to navigation.
- Section 3.4 details the implementation of the proposed gestural system using computer vision techniques.
- Section 3.5 describes a comparative user study used to test the gestural system against existing devices.
- Section 3.6 presents the results of the user study.
- Section 3.7 draws conclusions from those results.

3.2 Limitations of Current Approaches to Navigation

3.2.1 3D Mouse

.

Figure 10: The SpaceNavigator 3D mouse and its six simultaneous degrees of freedom

3.3 Gesture

While a gestural input system is the intended goal of this chapter, it is not unique to this research; researchers have been working on adding gestural interaction to GIS for many years. However interesting they may be, many of these have had major flaws that prevent ideal natural interaction, such as focussing quite heavily on discrete (rather than continuous) actions, or unnecessarily dividing up potentially simultaneous actions and assigning them to arbitrary hand configurations, for example Daiber et al. (2009).

3.4 Prototype System

To assess the capability for a direct gestural approach to improve navigation with a virtual globe, an early prototype gesture-based navigation system was developed. The system was set up to interpret a limited form of one-arm interaction into navigation in Google Earth.

3.4.1 Hardware

One of the main criteria was that the interaction be freehand — that is, the user would not be encumbered by any device. To facilitate this requirement it was necessary to use an external video solution to track the user's hand. A ceiling-mounted Point Grey Bumblebee2 stereo camera (Figure 11) was chosen and its performance was verified (see Figure 12 and Appendix A - Bumblebee2 Benchmarking Report). As with the Kinect, it provides both colour and depth data. Google Earth was displayed on a large screen rear-projection display (2.44 by 1.83 metres, illustrated in Figure 13).

Figure 11: Point Grey's Bumblebee2 stereo camera

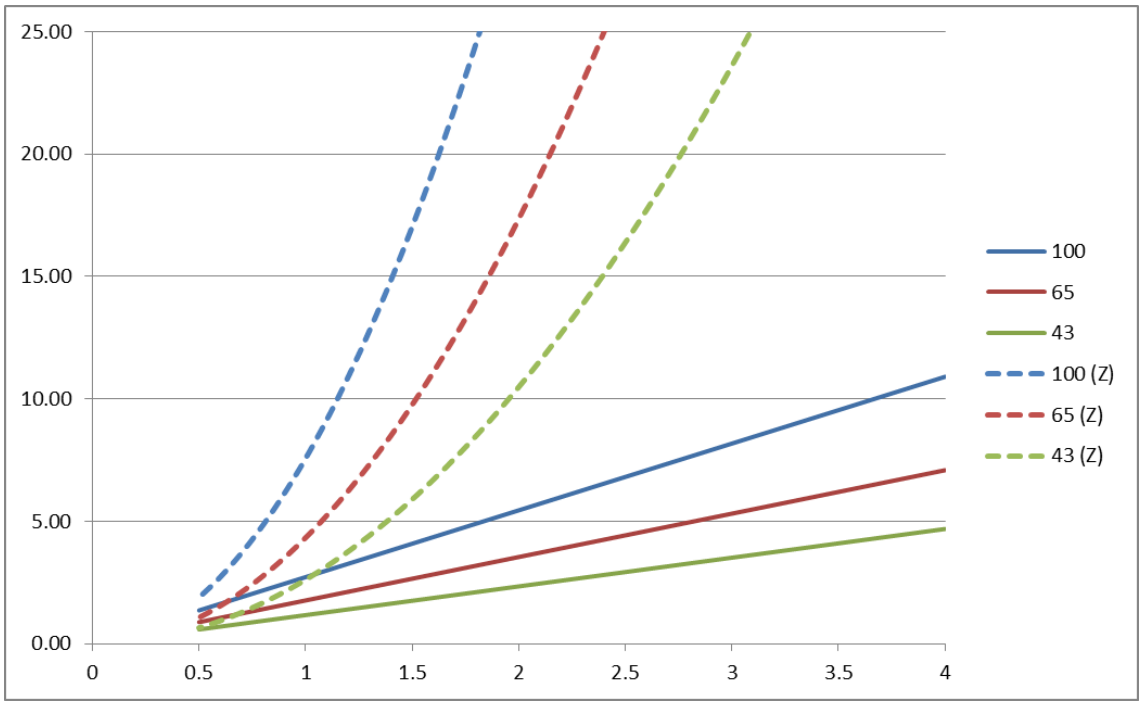

Figure 12: The spatial precision of the three different FoV variants of the 640x480 colour Bumblebee2 stereo camera — Each colour represents a different Field of View (right legend, in degrees). The solid lines represent the XY precision (vertical axis in millimetres) of an object at a certain distance (horizontal axis, metres). The dashed lines represent the precision of the depth estimate (Z ordinate) from the stereo algorithm at the certain distance.

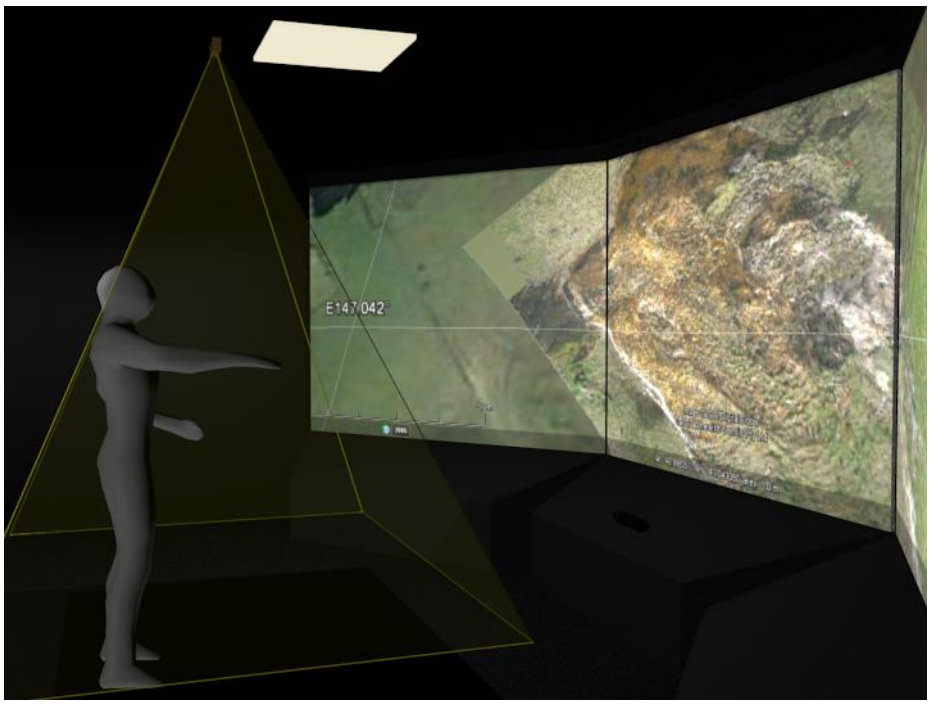

Figure 13: A model of the interaction environment showing the approximate position of the user relative to the screens and stereo camera viewing frustum — Ultimately only the centre screen was used.

3.4.2 Software

. For details on the rationale for this choice of implementation environment, refer to Appendix B - Platform Evaluation Report.

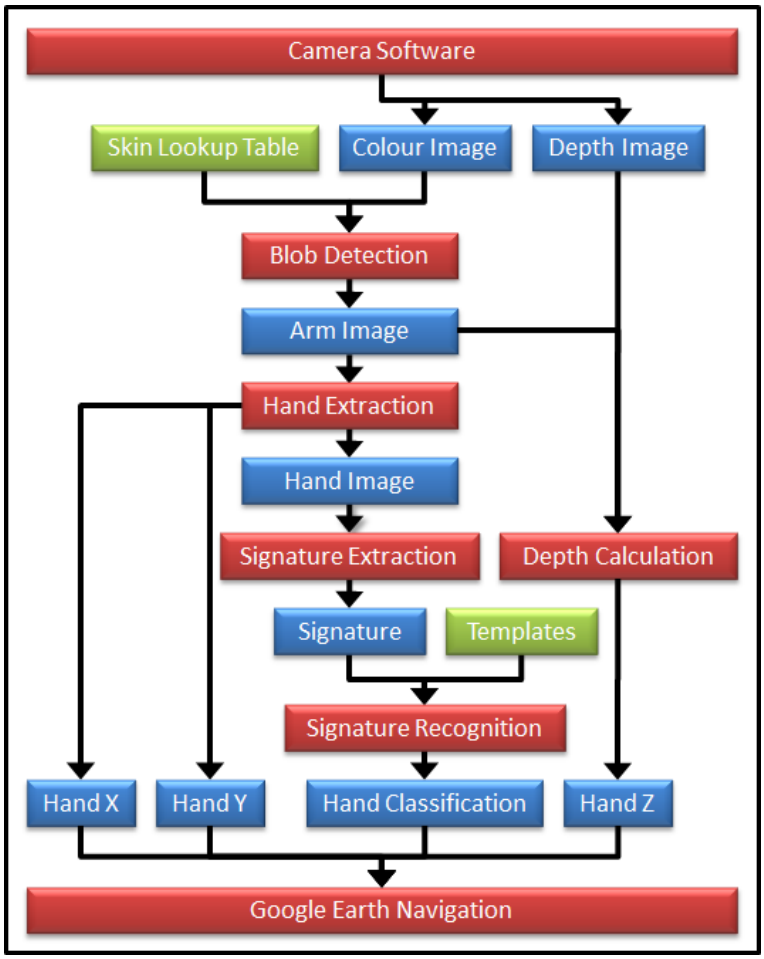

Figure 14: A diagram of the flow of data in the gestural navigation program — The blue boxes are the data produced by the various stages of the software (red). Green boxes represents precalculated data.

A custom block was written as an interface to the Bumblebee2 camera and was the source of all user input. It consisted mainly of calls to functions in the SDK developed by Point Grey for capturing and rectifying the images before running the stereo correspondence algorithm. This block produced three images as output: the left and right colour images and the depth image. These and the subsequent processes and forms of data are shown in Figure 14.

3.4.2.1 Segmentation

Then this image was converted to an array representing the width of the hand for each column of pixels. The array was then filtered by averaging the values in a moving window (± 10 columns). A simple heuristic was used to estimate where the hand started (Figure 16) by looking for a sudden jump in in these widths:

```
LET start = last_pixel_index – ENDING_LIMIT
FOR p = start TO last_pixel_index
      LET next_index = p + NEXT_SAMPLE_DISTANCE;
      IF next_index >= length THEN
            next_index = length - 1
      END IF

      LET current_width = widths[p]
      LET next_width = widths[next_index]

      IF current_width < MAX_SAMPLE_WIDTH
        AND current_width > MIN_SAMPLE_WIDTH
        AND next_width > MIN_NEXT_SAMPLE_WIDTH
        AND next_width - current_width >= MIN_SAMPLE_JUMP THEN
            LET cutoff = p - STARTING_BUFFER
      END IF
END FOR
```

The values for the constants were determined by trial and error on similar training images. Since the input data was a 2D image that had not been scaled, the apparent size depended on the distance from the camera. Since this information was available through the depth image (see Section 3.4.2.4), it was trivial to scale these values inversely proportionally to the depth value of the hand. The base values were as follows:

```
FILTER_RADIUS = 10
ENDING_LIMIT = 120
NEXT_SAMPLE_DISTANCE = 12
MIN_SAMPLE_WIDTH = 18
MAX_SAMPLE_WIDTH = 38
MIN_NEXT_SAMPLE_WIDTH = 20
MIN_SAMPLE_JUMP = 3
STARTING_BUFFER = 6
```

This cut-off point was then used to extract an image of just the hand part of the arm and also calculate the position of the hand in the original coordinate system (before arm normalisation).

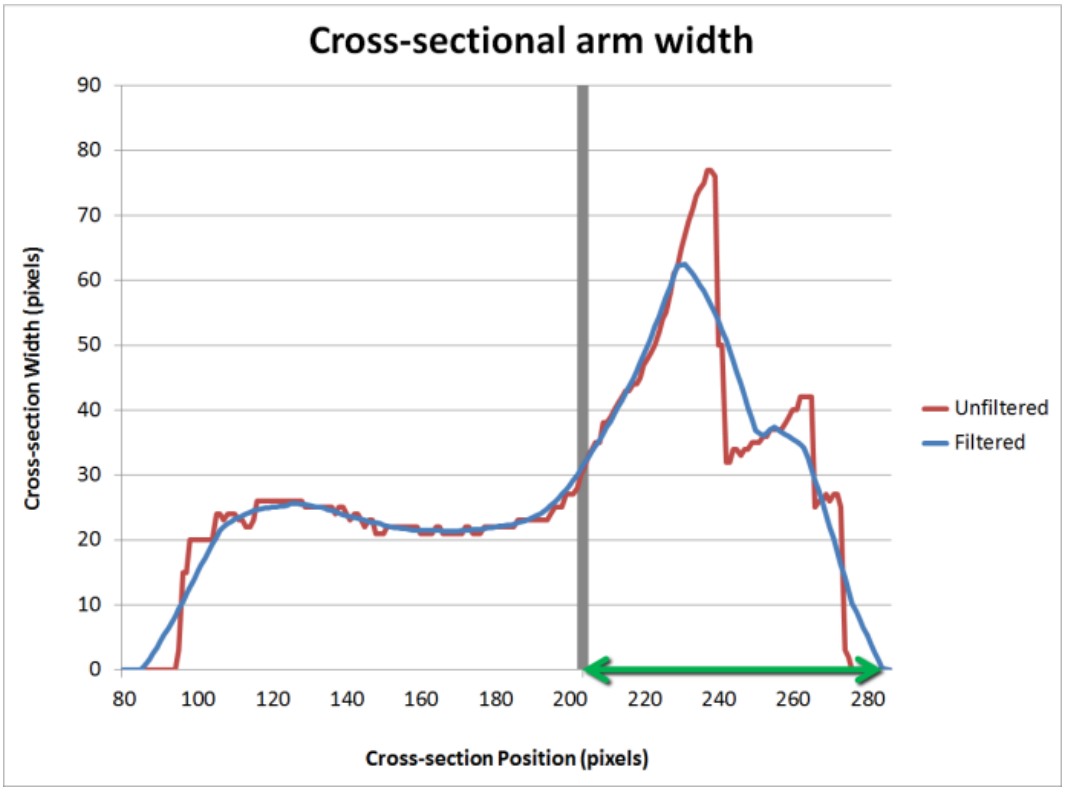

Figure 16: Example graph of the cross-sectional width of an arm showing the point where the wrist is detected

3.4.2.3 Hand Pose Classification

The state of the hand was recognised by first extracting a contour-based signature from the hand image and comparing it to labelled pre-recorded templates of different poses using a Dynamic Time Warping (DTW) metric, building on the work of Santosh (2010).

For both the pre-recorded templates and those captured in real-time, the contour of the hand was first extracted and normalised to a series of 100 equidistant points. For each of these points, the distance from the centroid of the hand was calculated and stored at the respective index in the signature array. On every frame during operation, the DTW algorithm was run to find the distance metric to the captured signature for each signature template in the library (Figure 20). The hand was then classified as having the pose corresponding to the template with the lowest difference.

The advantage of using DTW with such a contour-based classification approach was that subsections of the shape being matched are reasonably invariant to changes in other subsections, e.g., even if one digit is hidden, the algorithm is able to account for the shift in the position of the other digits in the signature array.

The advantage of using distances for comparison instead of Cartesian coordinates directly is that it makes the signatures invariant to rotation around the centre of the hand, as long as the signature in question and the templates it is compared to share a known starting point. Santosh's approach did not attempt to meet that requirement, instead comparing against 72 versions of the template, each rotated 5 degrees from the next. The novel advantage of normalising the orientation (as well as scale) of the larger arm object before slicing the hand at a consistent position is that it obviates the need to use such a brute force approach, thus making the classification process considerably faster.

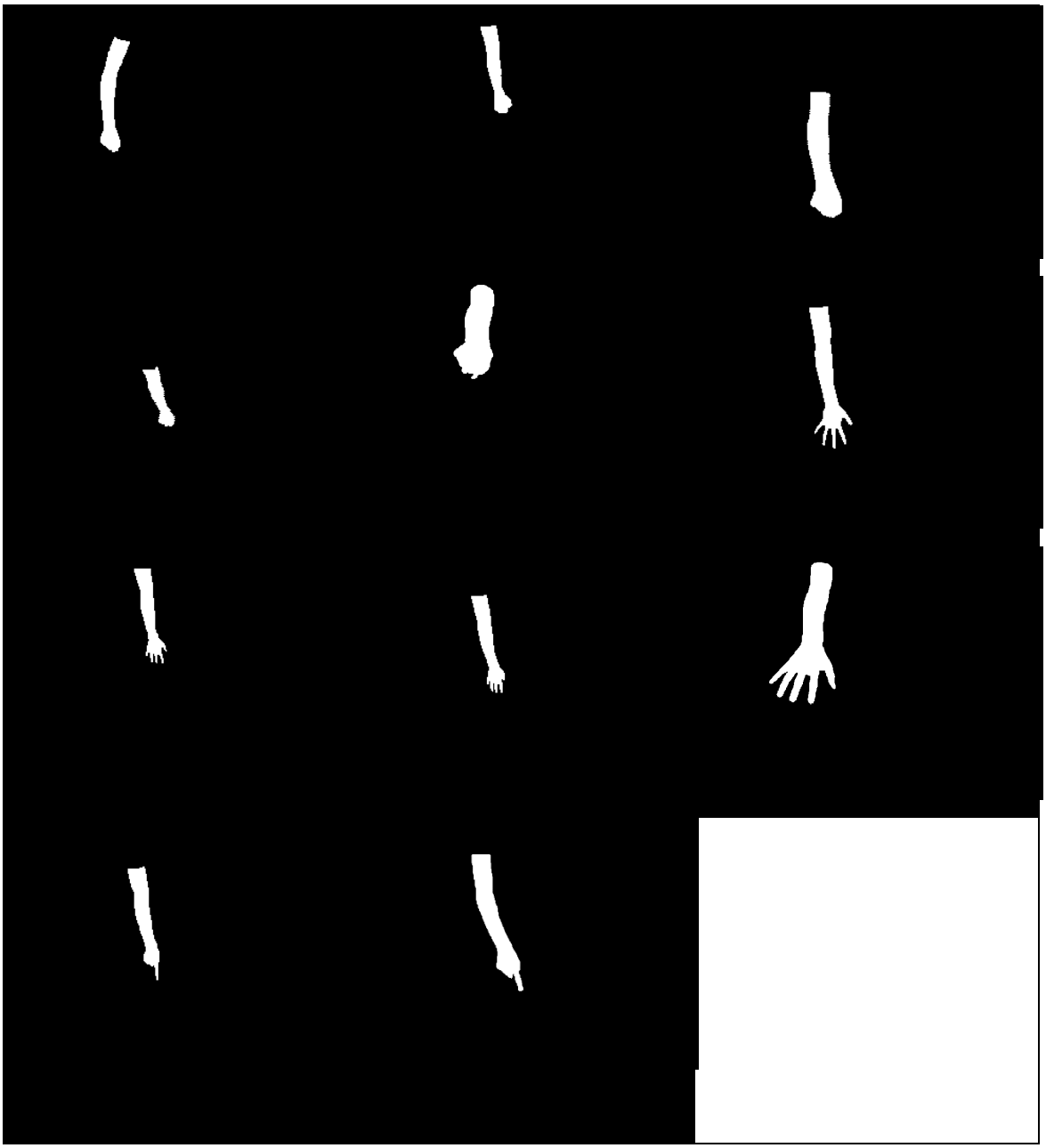

Figure 17: The template signatures used for classification: 5 *fist* templates, 4 *open* templates and 2 *pointing* templates

The signature template library that was compiled consisted of samples for each of three different poses: *fist*, *open* and *pointing*. Just a few samples (shown in Figure 17) from each were sufficient to provide reasonably accurate classification.

3.4.2.4 Spatial Coordinate Calculation

While these methods proved sufficiently robust at isolating and classifying the hand, there was no way to reliably track a given point on the hand and the noisy nature of the depth image meant that no single depth pixel in the could be relied upon to give a precise estimate of the hand's position. Instead, averages of both the XY and Z coordinates were used. In the case of the Z coordinates, the hand segment was applied as a mask over the depth image to limit the number of extraneous pixels entering the equation.

Since generating the depth image from the two colour images was the most CPU-intensive part of the system, special attention was paid to optimising it. Various permutations of the algorithm's parameters were tested until a suitable compromise between speed and precision was found. Furthermore, since the depth information was only needed to determine the position of the hand, the algorithm was only run on a defined region of interest of 160x160 pixels centred on the XY position already determined from the left colour image, considerably reducing the length of CPU time it needed.

3.4.2.5 Navigation

This equated to panning for the X and Y dimensions (the plane parallel to the display surface) and zooming for the Z dimension (perpendicular to the display).

3.4.2.6 Issues and Compromises

During preliminary testing a number of issues were encountered. First of all, there was a sizable amount of noise in the tracking. To combat this, it was necessary to reduce the sensitivity of the panning to reduce the noticeability of the problem on the display.

Another problem was the extent of inadvertent zooming while panning. It appeared that, due to the low sensitivity, users were likely to swing their arm radially to cover the distances required to pan. This led to motion along the Z-axis, as illustrated in Figure 18, which was interpreted as zooming by the system. To remove this problem, it was modified to treat panning and zooming as separate modes that could not be performed simultaneously. The mode was not determined until post-grab motion exceeded a certain threshold in any of the three primary axes, at which point a sound was played and all further motion during that clutching cycle was constrained to that plane (in the case of panning) or axis (for zoom).

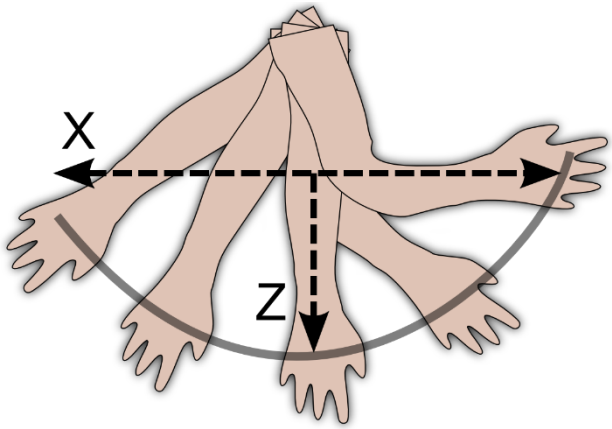

Figure 18: An illustration of the inadvertent motion in the Z axis that was often encountered while panning

Another issue was the unintended bias that the hand pose put on the spatial coordinates of the hand. As the open hand closed into the fist, the fingers largely disappeared under the palm, causing the average XY position of the observable hand pixels to shift closer to wrist. This was noticeable as a shift toward the user from the time the closing hand registered as the fist pose until the time that it was fully contracted. The equivalent opposite effect happened during the change back to the open pose. This change was big enough to negatively affect interaction. The XY coordinate averaging was changed to a fixed offset from the wrist, which quite successfully addressed the problem since the range of rotation through that plane (assuming the likely palm-down orientation of the hand) with respect to the direction of the arm at the wrist is very limited.

During initial pilot testing the variety of skin colours presented problems, but this was ironed out by adding more training images on which to base the set of Gaussian models. However, differences in the shape of people's hands still caused some problems in both the hand extraction and signature recognition components.

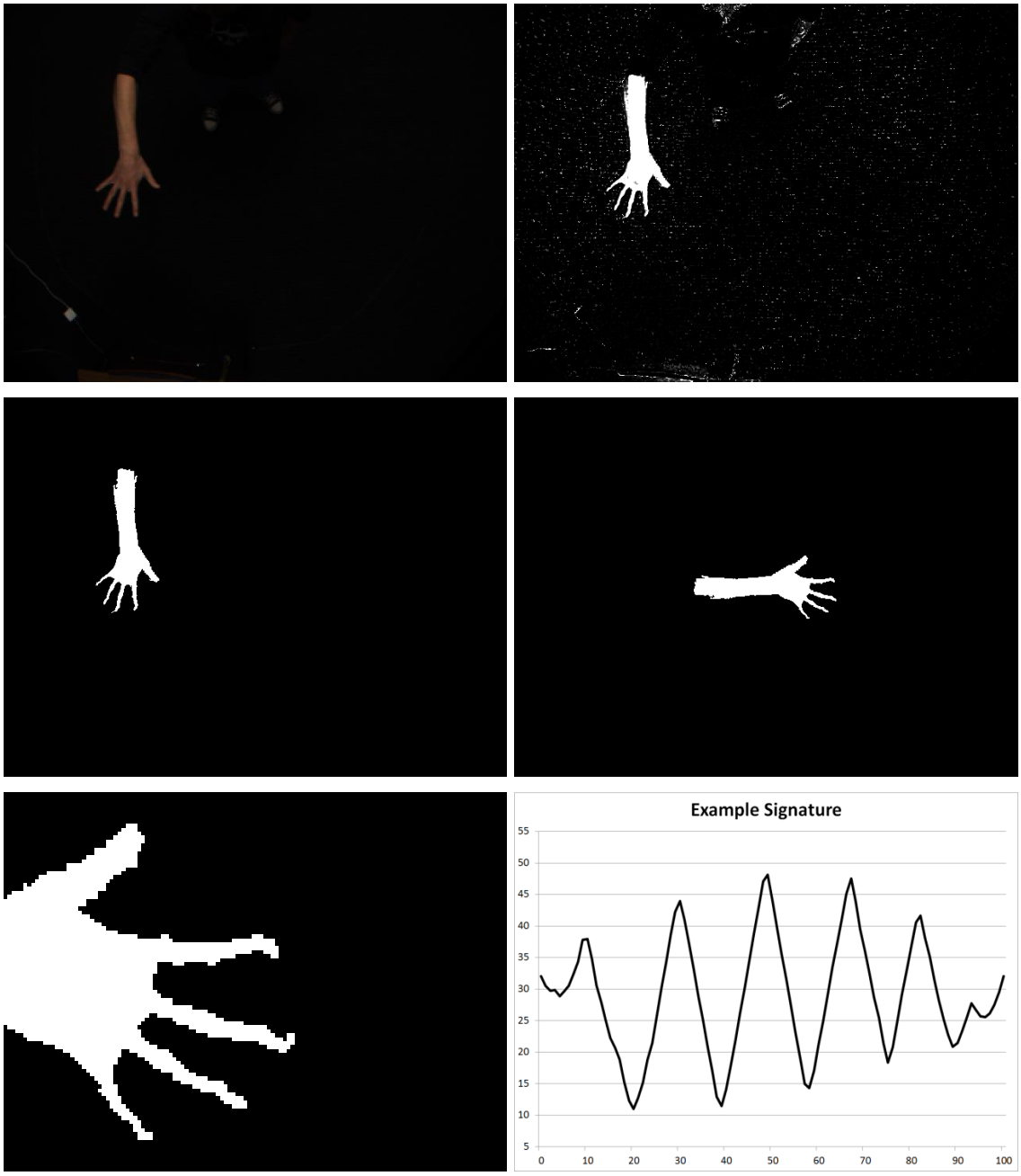

Figure 19: The steps in the process of obtaining hand shape signatures, showing the original frame (top left), the estimated skin pixels (top right), the arm blob segmented (middle left) and normalised (middle right), the extracted hand image (bottom left) and the resulting signature (bottom right)

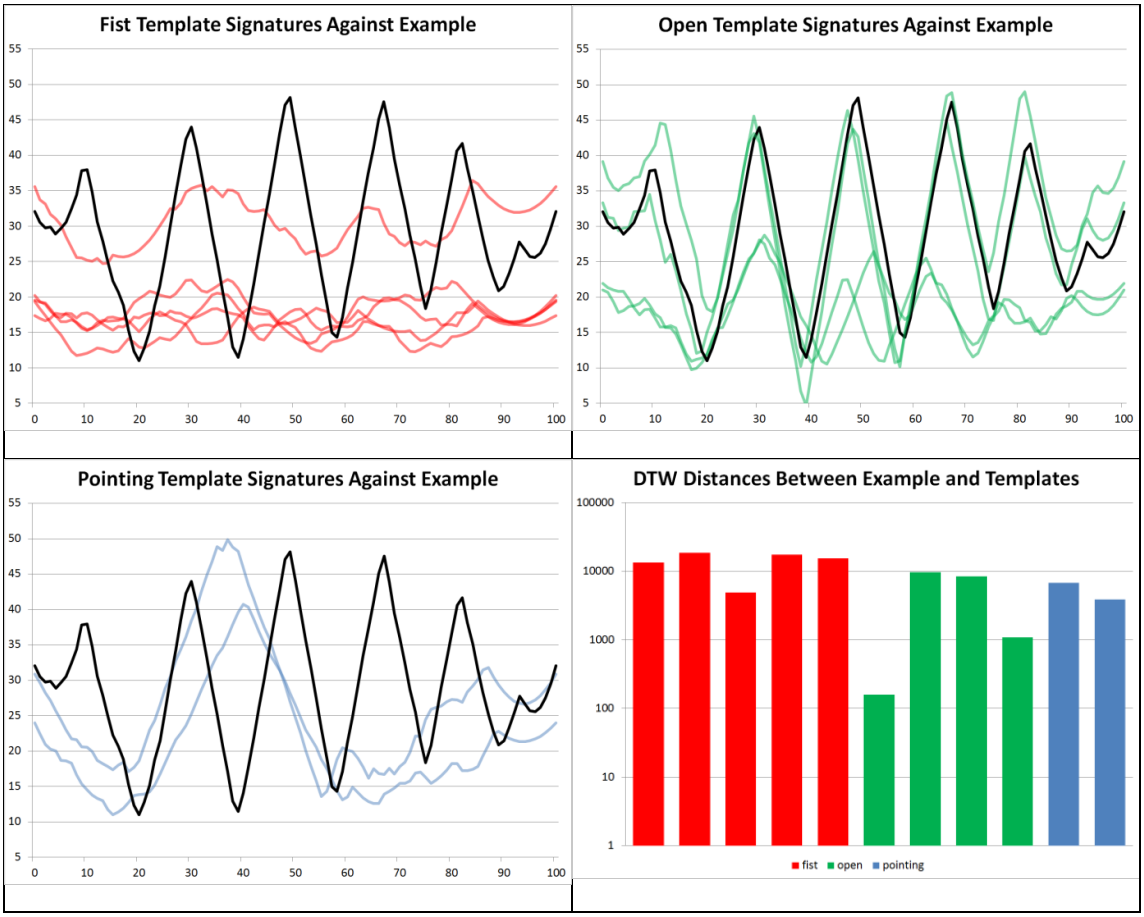

Figure 20: The extracted signature compared to the three classes of pre-recorded samples and a logarithmic graph of their similarity

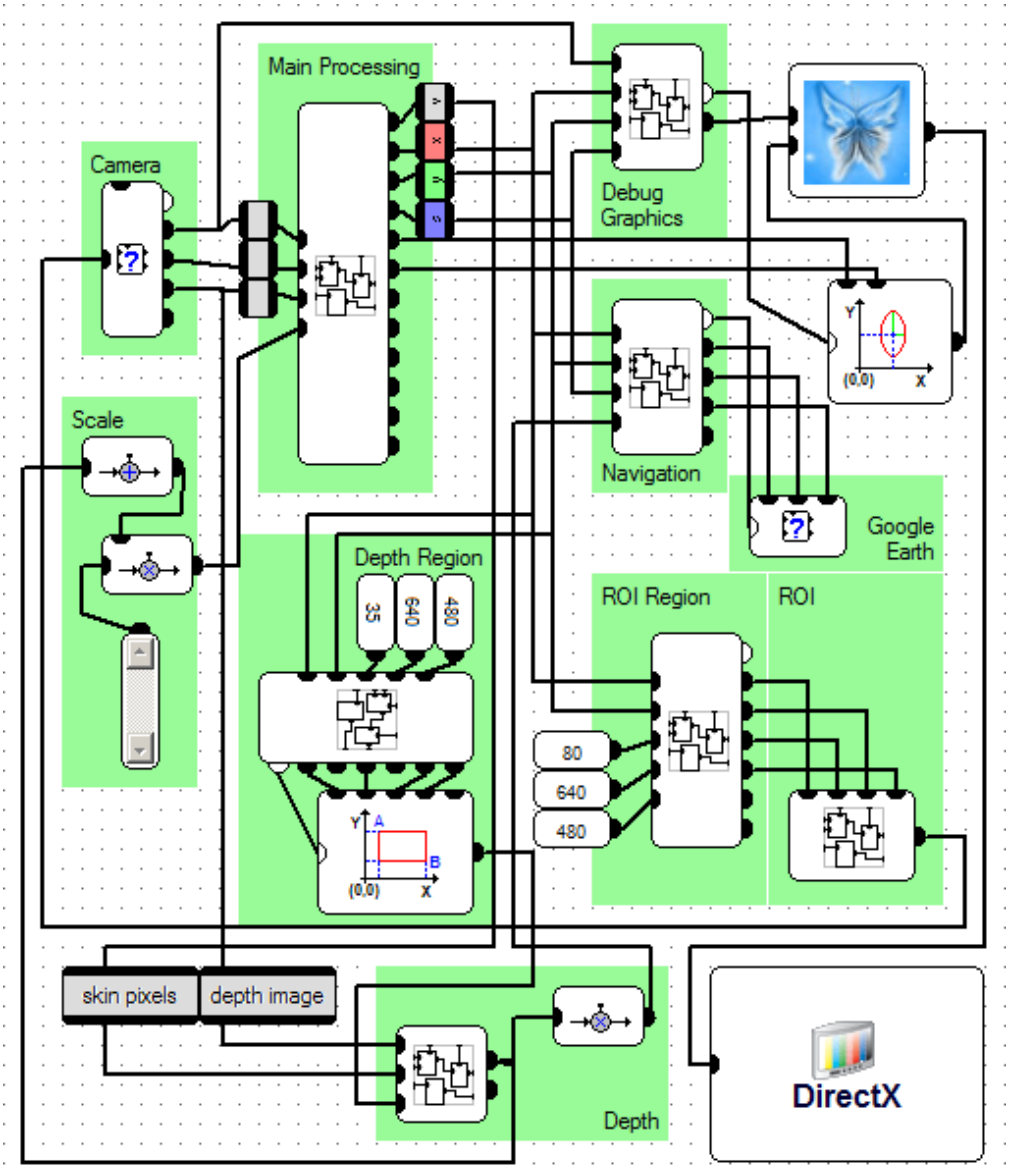

Figure 21: The top level arrangement of blocks in the EyesWeb patch — Most of the processing is done in the *Main Processing* sub-patch, which uses the input from the camera block to produce the <x, y> coordinates (red and green) as well as the state classification (blue) of the hand. The ultimate product of the entire process are the latitude, longitude and altitude parameters that are fed into the block controlling Google Earth.

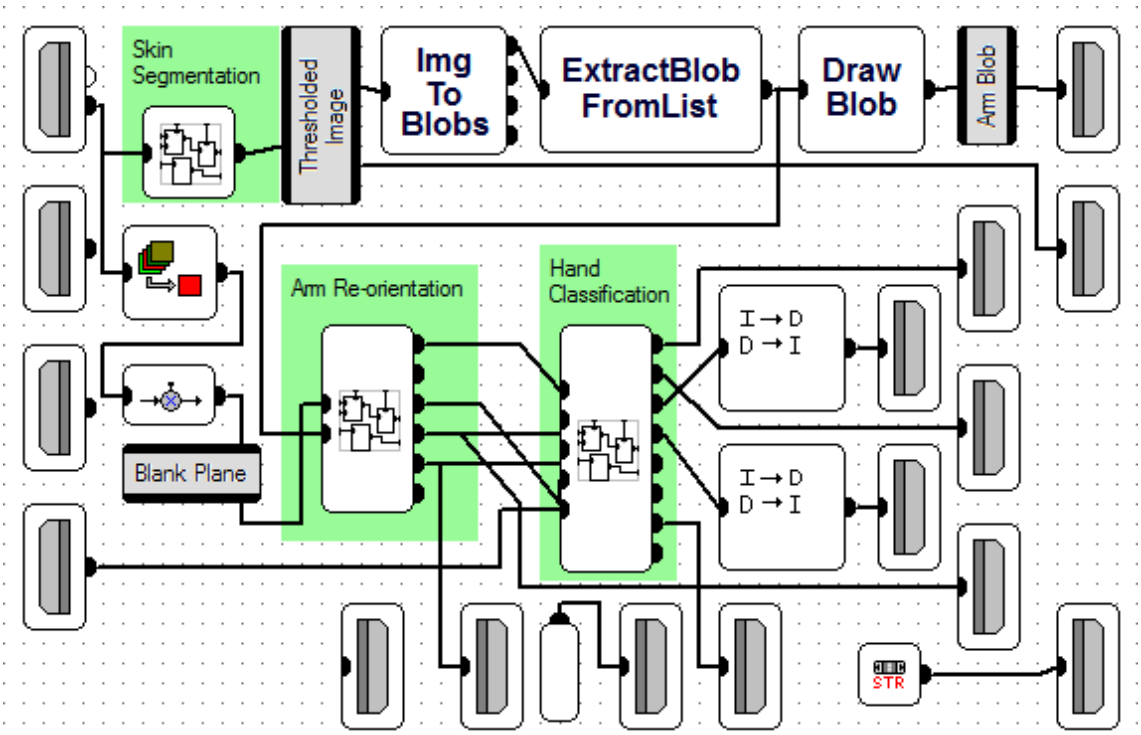

Figure 22: The *Main Processing* sub-patch — The colour input image is converted to a greyscale, which is thresholded into a black-and-white image from which the arm blob is extracted, normalised, processed and classified to determine the hand state and position.

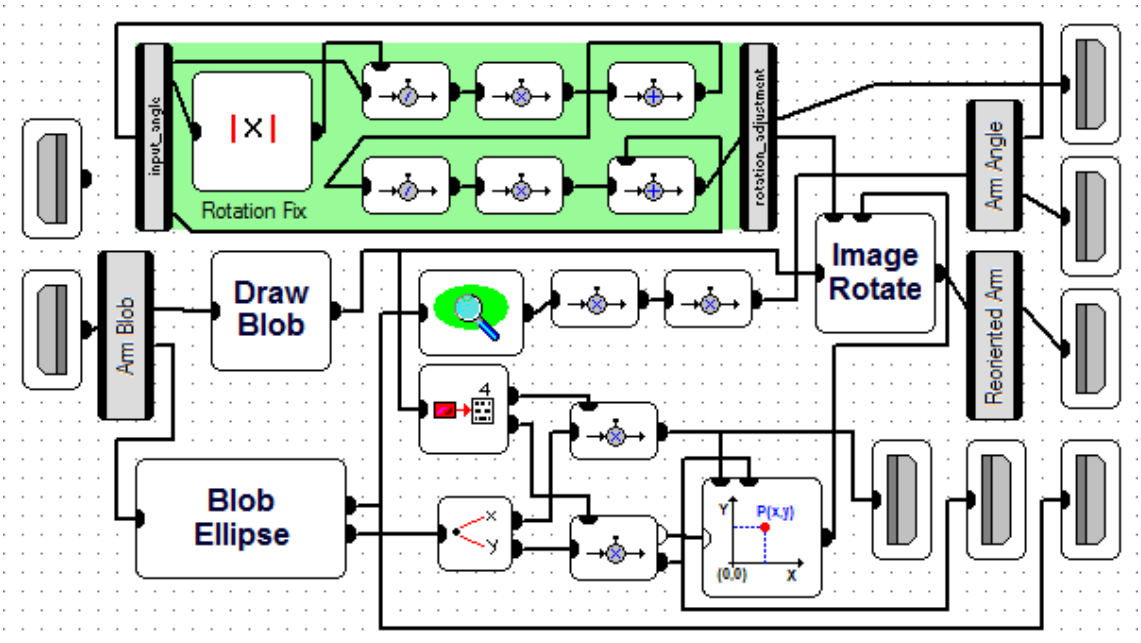

Figure 23: The *Arm Re-orientation* sub-patch

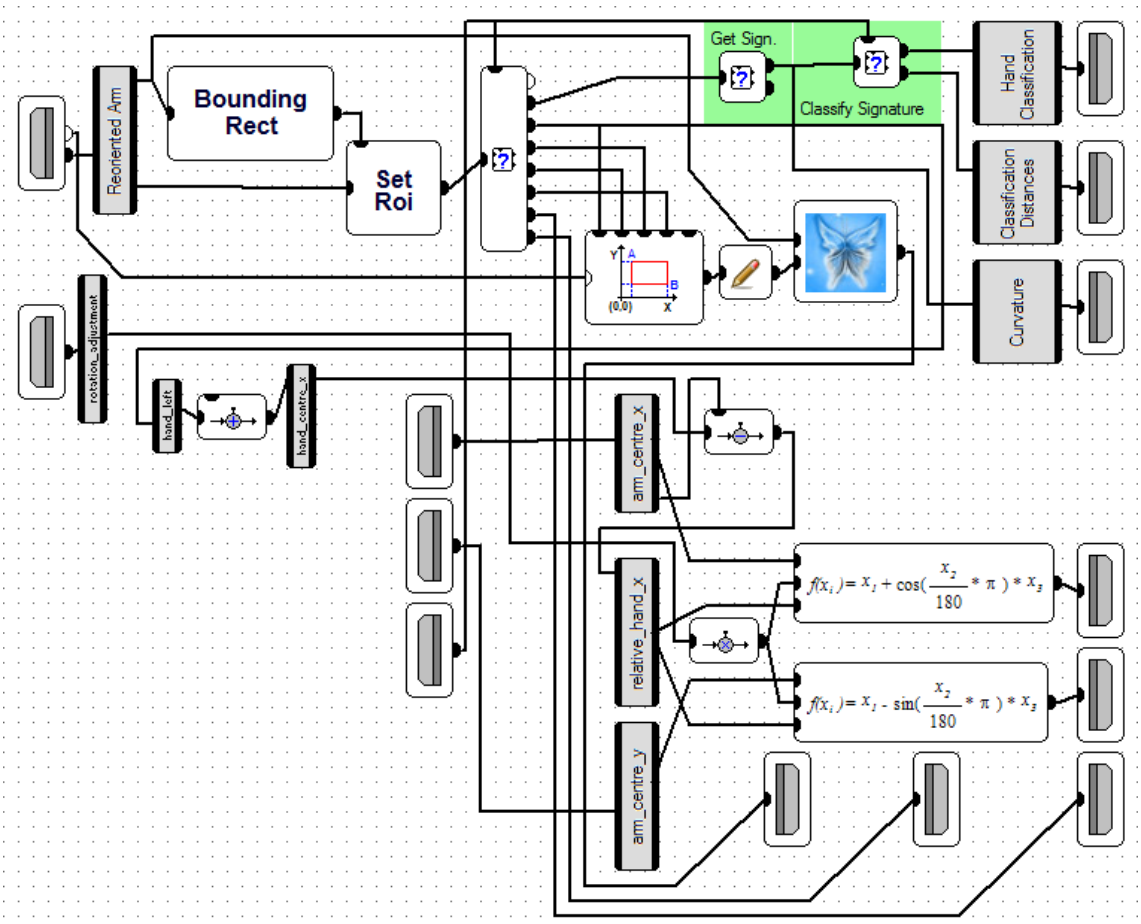

Figure 24: The *Hand Classification* sub-patch — The hand image is cropped out and the *Get Signature* block converts it to a signature. The *Classify Signature* block then matches it against a list of templates to produce the hand pose classification.

3.5 Experiment Design


1
2
3
4

3.6 Results

Parts of chapter 4 have been removed for copyright or proprietary reasons.

Parts of chapter 4 are based on the following paper: Stannus, S., Lucieer, A., Fu, W. (2014). Natural 7DoF navigation & interaction in 3D geovisualisations. In Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology - VRST '14. ACM Press, pp. 229-230.

Other parts of chapter 4 have also been removed for copyright or proprietary reasons.

Parts of chapter 4 are based on the following paper: Stannus, S., Fu, W., Lucieer, A., (2014). Natural 7DoF input for 3D navigation. In Proceedings of the 26th Australian Computer-Human Interaction Conference - OzCHI '14. ACM Press, pp. 216-219.

Chapter 4 - A Pose Model for 7DoF Navigation and Interaction

It suddenly struck me that that tiny pea, pretty and blue, was the Earth. I put up my thumb and shut one eye, and my thumb blotted out the planet Earth. I didn't feel like a giant. I felt very, very small.

— Neil Armstrong

4.1 Introduction

This chapter covers the development of a more effective design for natural navigation, dubbed *AeroSpace*, inspired by the lessons learnt from the user study outlined in the previous chapter and guided by more careful consideration of the theory-based design goals. In particular, the aim was to design a technique that was bimanual and consisted of a sufficient number of integrally simultaneous degrees of freedom. The other chief focus was to improve the tracking implementation to the point that imprecision no longer detracted from its effectiveness. The chapter is structured as follows:

- Section 4.2 reassesses the problem of navigation in greater depth and tracks the design process that culminated in the *AeroSpace* technique.
- Section 4.3 describes the comparative user study designed to test its practical validity and the details of the hardware and software implementation of *AeroSpace* that was used.
- Section 4.4 reports the results of the user study.
- Section 4.5 discusses the conclusions that were drawn from this work.

4.2 Revisiting Navigation

While the results of the first user study are useful in terms of improving navigation performance, they do not speak to the validity of the problem definition itself. Section 4.2.1 attempts to give the ideal such definition by examining navigation in its purest form and deducing a series of consequent abstractions with which to augment it. Section 4.2.2 rigorously narrows the scope of possible interaction techniques by evaluating the theoretical constraints to arrive at the fundamental metaphor most suited to this problem. Existing literature is then reviewed in Section 4.2.3 to outline in which ways previous interaction techniques have resembled this metaphor and in which ways they have not. Section 4.2.4 then proposes a practical extended mapping of this metaphor to the constraints of human hands.

4.2.1 Defining the Problem

While mimicking the real world in the ways outlined in Section 2.4 is a useful aim, a perfect simulacrum of the world itself is insufficient. Obviously the pursuit of natural interaction has to be balanced with the implied goal of taking advantage of some magical (Kulik 2009) properties that only virtual representations can allow.

The first such unnatural requirement is to augment the real world with abstract data. Augmented reality meets this, but ideally the user should be able to interact with the data from any point of view he/she wants, even remote islands or outer space, without the inconvenience of travelling there. Desktop geovisualisation removes this restriction by breaking the natural mapping between real-world and *virtual navigation*, but if one imagines a static two-dimensional virtual world displayed on a large high-resolution wall-mounted screen, this is a property that emerges naturally, since the user's view of any point in the virtual world is relative to where they are standing. Ball et al. (2007) showed that users prefer such *physical navigation* (pictured in Figure 27) and conjectured that it is generally a more efficient method than virtual navigation.

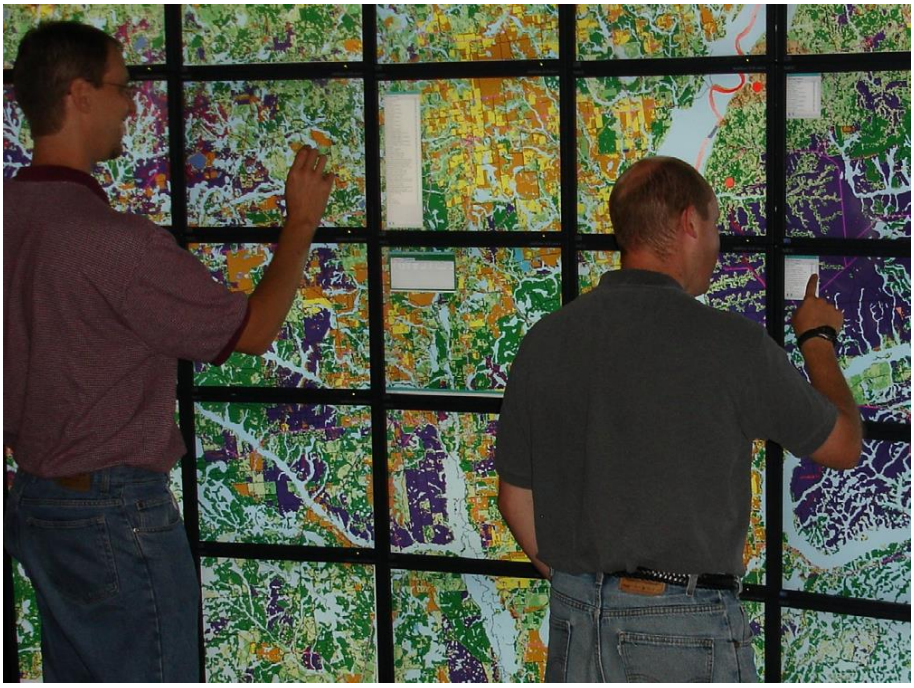

Figure 27: An example from Ball et al. of physical navigation of a 2D geovisualisation

Physical navigation in 3D environments requires rendering according to a tracked head position. Ideally this head-tracking is modelled for both of the user's eyes and presented stereoscopically, resulting in less user errors and faster performance than no head-tracking (Ragan et al. 2013). However, even if these criteria are met, there is still the substantial problem of occlusion of objects *in front of* the screen. Head mounted displays are one way of avoiding such occlusion, but they are not without their own issues, such as limited resolution and user encumbrance. Another way of getting around occlusion is to use a half-silvered display (e.g., in the manner of Hilliges et al. (2012)). However such displays usually allow only a small interaction space, especially constrained by the reflecting layer.

However, there are practical limits to physical navigation, since it is constrained by both the distance that the users are willing to cover and the size of the displays. Because of this, *virtual navigation* that allows the *positional* offset between virtual and real world to be adjusted is still necessary. *Orientation* also needs to be relative, as not all real-world directions can be viewed in most display environments. For example, the large-screen environment described in Section 3.4.1 gives the user a virtual field of view of approximately 180 degrees horizontally and 90 degrees vertically, so if only position is virtually controllable then it would only be capable of displaying a minor subset of possible lines of sight from any given position. Furthermore, even with full 360° vision, some angles, such as straight up or straight down, are not ideal for viewing.

While a 1:1 correspondence in scale between physical and virtual space seems logical, it would be unhelpful when the data the user is interested in covers ranges of anything more than a few metres; in this situation, the viewing angle becomes invariant to physical navigation, data moves far outside the user's reach and depth perception loses its power (Cutting 1997). However, these problems disappear if the user is able to adjust this scale ratio as needed (Wartell et al. 1999). This is not the same as adjusting the control-display ratio, since the relationship between those would still be 1:1; instead it would be the ratio between the display and the virtual environment that would change. This would be the three-dimensional equivalent of zooming.

Since the virtual world is three-dimensional, the position and orientation of the virtual camera can each be represented as three-dimensional vectors. The scale ratio can be represented as a scalar value. Therefore full navigation in a 3D geographical environment should be represented as the manipulation of a 7-dimensional state (as pictured in Figure 28).

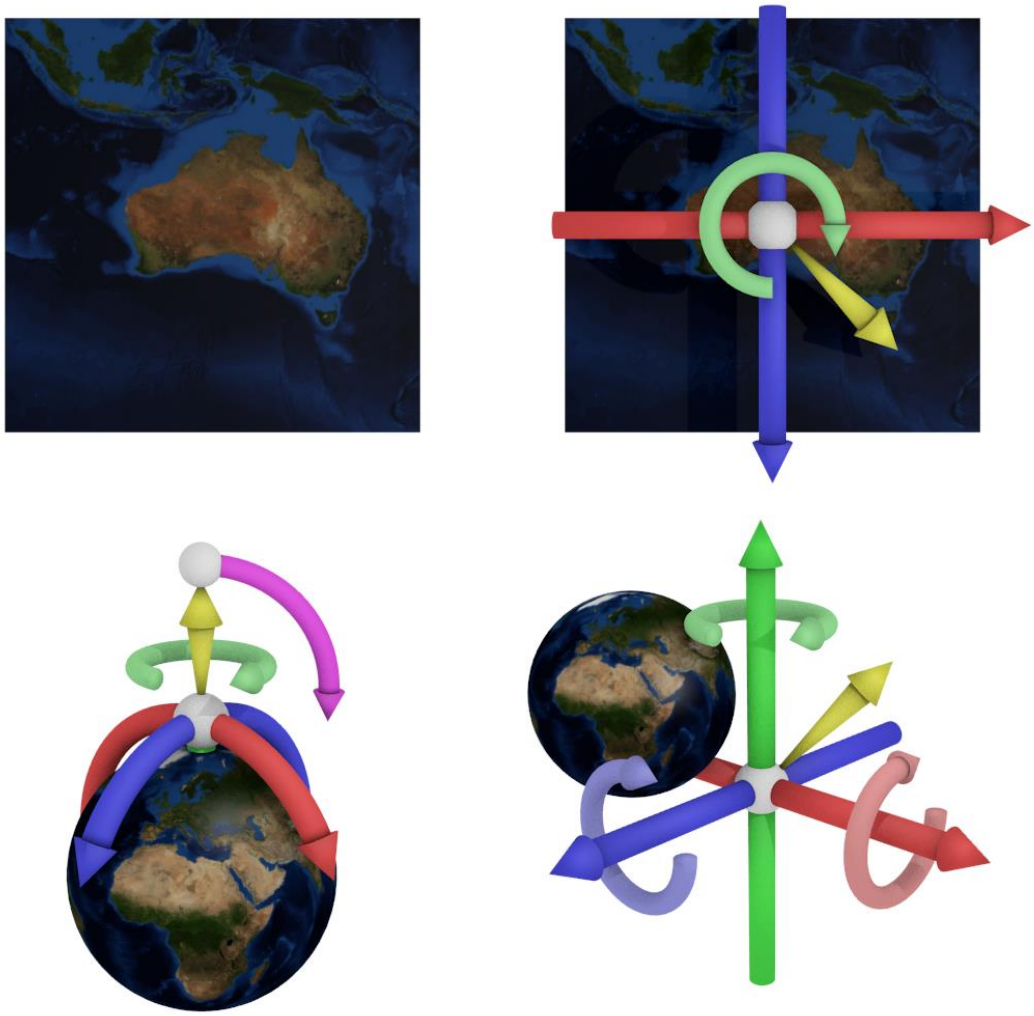

Figure 28: Four different models for navigation — The first image (top left) represents traditional 2D map visualisation, in which there is no virtual navigation. On the other hand, even 2D versions of electronic maps can have up to 4 degrees of freedom (top right). Blue and red represent the latitude and longitude respectively, which in basic 2DoF navigation is usually aligned directly to the X and Y display axes. The light green and yellow arrows represents the roll and zoom degrees of freedom which are used ubiquitously on touch devices. Geovisualisation environments like Google Earth treat the world as a 3D environment (bottom left) and in some cases add a further degree of freedom in the form of tilt (magenta) but still constrain the camera to face the surface. The ideal model (bottom right) allows the camera to be positioned anywhere, with any rotation and replaces zoom in terms of elevation or field of view with the separate concept of scale.

4.2.2 Designing a Navigation Metaphor

To maximise efficiency, the interaction technique used for navigation should allow as many as possible of these 7 degrees of freedom to be performed simultaneously, in a direct and suitably integral manner. The obvious approach would be to use a 6DoF device in one hand that maps the 3 degrees of freedom of position directly to translation, and those of orientation to rotation, but this would still leave one degree of freedom unaccounted for. This could be implemented using some kind of joystick or buttons on the device, but these would be indirect solutions.

4.2.3 Previous Work

Zeleznik et al. (1997) implemented a somewhat limited method for interaction in 3D virtual space using two 2DoF controllers, whereby the 2D clutch points were mapped to 3D by picking the points they collided with in the projective space. They argued that such techniques should be based on physical analogues and in general two-cursor techniques are more appealing to users than single-cursor ones.

4.2.4 Proposed Navigation Design

This section describes *AeroSpace*, a novel interaction method that implements the two-point method in 3D for spatial navigation. The input requirements are minimalist, allowing use with a wide range of hardware. The core design requires positional tracking of the user's hands and head in 3D space. The head-tracked display must also be stereoscopic, as the user's understanding of scale will be fundamentally necessary (Bowman et al. 2004).

In the natural world manual interaction occurs with surfaces, not volumes or arbitrary points therein, so there is no perfectly natural action that can be replicated to control

clutching. However, pinching with thumb and finger is close enough a metaphor for purposes of this interaction and maintains the haptic feedback of clutching. This feedback is a crucial part of real world interaction and as long as the contact can be detected accurately the user would possibly face less delay (El-Shimy et al. 2009) than if they had to wait for a visual or auditory cue. Furthermore, the positional displacement required to toggle between the clutch states can be minuscule such that even ignoring the feedback advantage, time is saved through much shorter finger paths. Less motion could also conceivably reduce unintended alteration of the hand position (of the type mentioned in Section 3.4.2.6) and the negative effects of repetitive strain.

To control the roll around the ambiguous axis, a novel solution would be to use the direction of the index finger on one of the user's hands in a pointing pose. One advantage is that it would be relatively easy to detect for most tracking systems; the tip of the finger can be used and rotation around the axis of the finger (pronation and supination of the forearm) can be ignored. Furthermore, assuming the user's hands will spend most of their time in front of them, palm down, roughly facing forward, the absolute axis around which it will be easiest for them to rotate their fingers will roughly correspond to the ambiguous rotation axis. This is because it will line up with the high ranges from maximum flexion to extension of both the wrist (105° (Grandjean 1982)) and metacarpophalangeal joints (90° (Huang 2000)) in the fingers. Together, this gives a range of at least 180°, enough to allow any relative rotation to be reached in one *move→clutch→move→unclutch* cycle. A further 180° is technically possible if flexion of the interphalangeal articulations (last two finger joints) is considered, though detection capabilities might be compromised in such situations.

Though this mapping precludes use of the index finger for pinching to control clutching, contact of the thumb with the side of the middle finger can be used instead. This would not only mimic how users might pose their hand during pointing (see Figure 30), but also allows for clear detection of the index finger by external visual tracking systems. This mapping is not perfect; it generalises the 3 rotational degrees of freedom of the finger to a single degree of freedom input and, as mentioned above, relies on the range of rotation of the user's hand and finger, which may cause minor strain or occasionally require more than one phase of clutching. However, it could be considered the most elegant of a limited set of design options.

- One hand pinching: panning.
- Two hands pinching: 6DoF navigation.
- One hand pinching, one pointing: 7DoF navigation.

During 6DoF navigation, the roll is fixed with respect to the plane formed by the origin and the start and finish *direction* vectors of the bar between the two hands.

This design keeps the implementation requirements to an absolute minimum while still allowing the user to use his/her preferred hand for the asymmetric interaction. It also leaves one combination (both hands pointing) free for cycling between modes, of which navigation might be just one (Figure 31).

Other methods, such as using movement or pointing direction while this combination is held to jump to specific modes might be more efficient if there are many modes to cycle through.

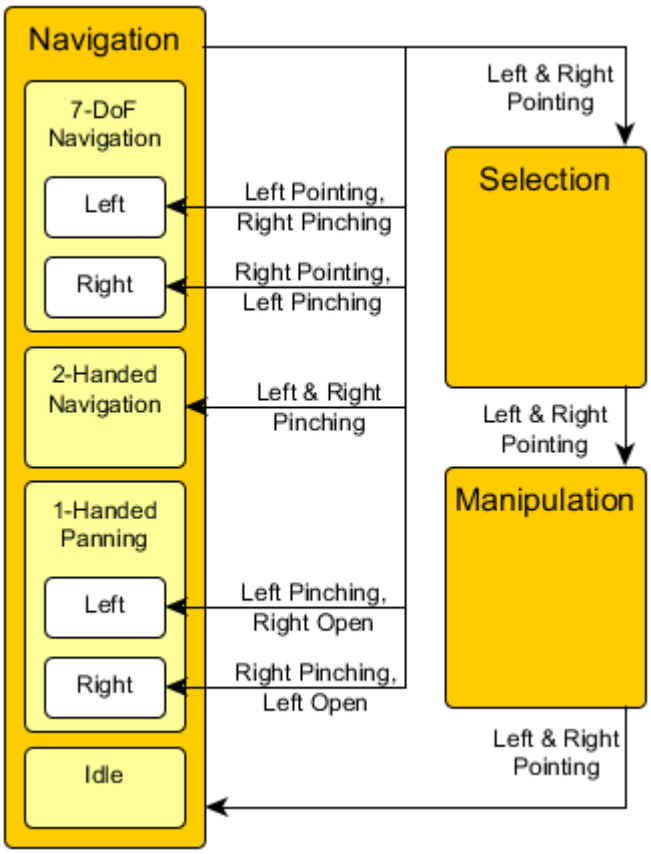

Figure 31: A state diagram of the proposed general modal approach to interaction, in which navigation is one of possibly any number of modes. Internally, navigation does not involve hysteresis; all its sub-states depend only on the current hand poses. Using the pointing pose with both hands at the same time cycles through the modes *à la* Windows' Alt-Tab.

4.3 Method

The *AeroSpace* design was then implemented and a comparative user study conducted to test it. The SpaceExplorer 3D mouse was chosen to compare against because it is a commercial device designed for geovisualisation and other 3D spatial interaction. Also, there were a number of differences between the two devices in terms of the design goals and interaction principles mentioned in Sections 2.3 and 2.4. The most obvious of these is that 3D mouse uses pure rate-controlled spatial input, via its isometric puck. Also, while the *AeroSpace* design is bimanual, the 3D mouse only uses one hand for spatial input. Though both the 3D mouse and the normal two-point mode of *AeroSpace* exhibit six simultaneous degrees of freedom, *AeroSpace*'s point-pinch mode reaches the full seven degrees of freedom.

The aim was to show that, due to these fundamental differences, the *AeroSpace* approach would outperform the 3D mouse with users who had no experience with either

device. It was hypothesised that *AeroSpace* would be perceived as more natural, because of both the directness and general form factor of the gloves. It was also expected that the lighter gloves and symmetrical bimanual approach of *AeroSpace* would be more comfortable. Because of these factors, it was expected to be faster as well.

The decision to not include a standard mouse in testing was made for a number of reasons. Unlike the first user study, the aim was to test interaction and 7DoF navigation in an embodied virtual environment. A mouse might cope with map and basic virtual globe interaction up to 3DoF, but there is no established way of fairly mapping it to 7DoF interaction. Using a mouse would have also severely restricted physical navigation by constraining participants to interacting at a desk. It was instead judged that the participant time would be better spent looking more in depth at the other two devices.

4.3.1 Hardware

The system was run in a large stereo display environment consisting of three screens (2.44 by 1.83 metres each, arranged in a concave fashion with the screens meeting at 45°; see Figure 32) lit by a total of 6 rear-projection projectors fitted with polarising filters. An ARTrack3 infrared tracking system with 3 cameras was used to track the six degrees of freedom of the user's stereoscopic glasses.

Though ideally the user should be as unencumbered as possible, the decision was made to use gloves (Figure 33) for tracking and pose detection since they were much easier to track accurately and precisely and were the only feasible way to detect contact. It would have been possible to detect approximate contact from hand pose using computer vision, but doing so would have meant a large reduction in classification accuracy; at the very least the haptic feedback would not have always coincided with the detected pose in the system. Each glove had three patches of conductive fabric wired back to a small re-purposed Bluetooth keyboard, which allowed the contact states to be detected as key-presses in software.

Each glove also had two positional markers, attached in alignment with the index finger, giving each hand five degrees of freedom: position and direction, but not full orientation, since the roll around the index finger axis was not known. Technically this configuration did not even provide direction, since the two markers were identical and there was no way for the system to know the sign of the ray formed by them. However, it could be reasonably assumed that all meaningful interactions that made use of this direction started with it facing in the more forward of the two possible directions, i.e., the one with the negative (screen-facing) Z component. In case the direction changed to be more toward the user during active interaction phases, the new direction was compared with the previous one each frame and the sign that resulted in the least rotational change was assumed correct.

The ARTrack system took care of processing the images from its infrared cameras and correlating the results into 3D coordinates, but it did so for each marker ball separately so it was still necessary to implement a way to deduce which two points correlated to which hand. This was solved by making the baseline physical distance between the two markers unique for each hand. Distances of 5 and 7cm (for the left and right hands

respectively) were chosen since they were different enough that the two pairs would not be confused for each other, long enough that the two markers of the shorter pair would not obscure or disrupt tracking of each other and short enough that the marker attachment with the longer pair was not too unwieldy. Fortunately, ARTrack automatically resolved the more complex 6DoF arrangement of markers on the stereoscopic glasses, so it was almost guaranteed that only the 4 marker points of the hands would be provided to the system. In most cases, picking any point and comparing its distance to the other 3 would clearly determine which hand it belonged to. Due to inaccuracies in measurement and possible warping of the marker attachments, it was necessary to allow a certain amount of tolerance in matching observed distances to the predefined values. However, this had to be weighed up against the possibility of erroneous matches when the hands would come close to each other. A value of ±10% of the baseline distance was determined to be a reasonable compromise.

.

4.3.2 Software

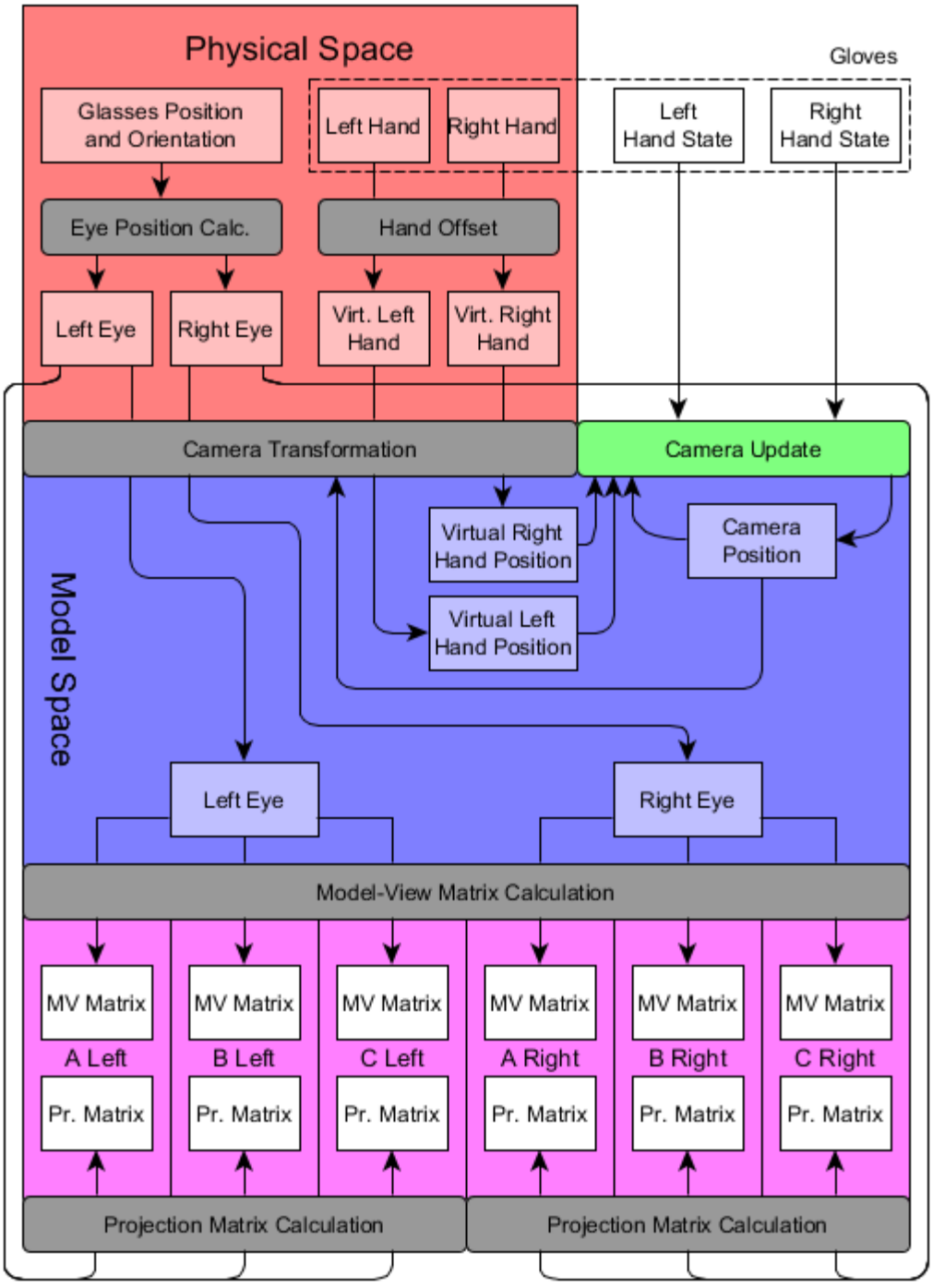

Figure 34: A diagram of the spatial variables (rectangles) existing in different spaces (enclosing boxes) and the processes (rounded rectangles) involved in their calculation — These processes ultimately result in the separate projection and model-view matrices for both polarities of the screens A, B and C.

Virtual representations of the user's hands were rendered in different colours. During pinching, a hand would become a simple sphere and during 7-DoF navigation, a line was rendered between the two hand points and a cone from the pointing hand in the direction of pointing (see Figure 35). Audio feedback was provided in the form of distinct grab and release sounds when pinches started and ended. A black fog was applied to the rendering to assist in the perception of distance.

3D mouse navigation was implemented by moving the virtual world in whatever direction the user pushed or pulled the puck, assuming it was held flat and facing forward (i.e., the same orientation as normal desktop use). The world was similarly rotated around whichever axis the puck was tilted or turned. The centre of rotation was a point just in front of the central screen. 3D crosshairs were rendered at this point to assist in visualisation. A separate pair of buttons (+ and −) could be held to adjust the scale at a constant geometric rate (±50% per second).

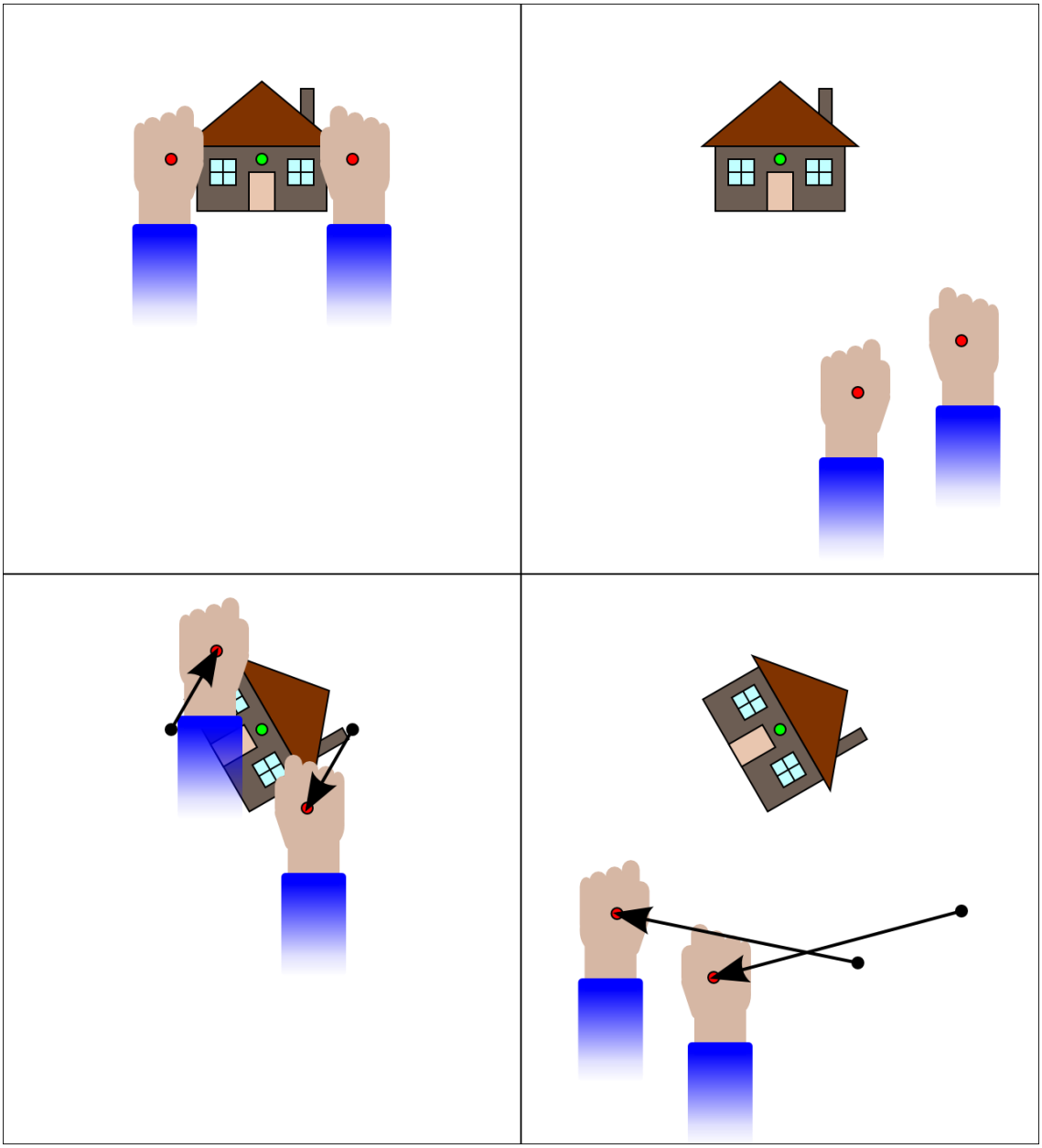

Figure 36: Two different ways of acheiving the same rotation transformation using the two-point approach — One approach (top-left becomes bottom-left) uses points close to the object that is to be rotated while the other (top-right becomes bottom-right) uses a more distant pair. While the first approach risks obscuring the focus of the user's transformation, the second approach must cover more ground (black arrows) to effect the same rotation without translating the object. This phenomenon extends to three-dimensions and suggests that users will keep their hands close the the objects of their attention to minimise physical exertion.

4.3.3 Procedure

The tests were intentionally based around real-world geospatial data, namely a digital elevation model and orthophoto of a landslip. This data was semi-automatically produced by Lucieer et al. (2014) from aerial photographs (of which Figure 37 is an example) captured by a small multirotor unmanned aerial vehicle (UAV).

Figure 37: An aerial photo of the landslip around which the experiment data was based

In order to properly compare the devices, tasks were chosen that are fundamentally spatial in nature and involved navigation as well as other forms of spatial interaction. Each type of task was performed once as a practice and then again but with more demanding goal data and the request that it be completed as quickly as possible.

The first type of task was pure navigation, requiring the participant to navigate such that the goal area ended up within a fixed space of the room. The purpose of this task was to test just the navigation aspect of geovisualisation. The practice goal was to place Australia within a frame more or less corresponding to the central screen. The timed goal required zooming in on a pinpoint to reach the landslip area and then placing it within the slanted 3D box depicted in Figure 38. Completion was signalled automatically when the physical→virtual transform was within a certain range (7.5cm physical positional, 7.5° rotational and 12.5% scale difference) of the predetermined solution.

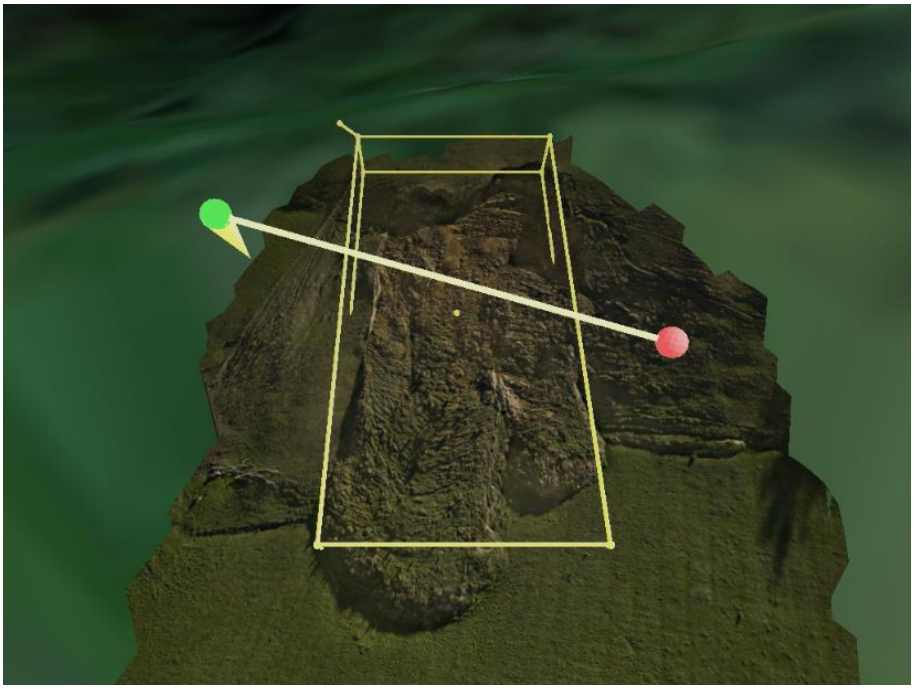

Figure 38: A screenshot of the Navigation task, which was the first of the three to be performed for each device. This image shows the landslip being placed roughly inside the static goal frame.

The second type of task required the participant to accurately mark several ground truth markers in the orthophoto (see Figure 39). This task was designed to emulate the process of manually marking GPS ground control points in the initially-generate model to position it geographically (Lucieer et al. 2014). The goal in the practice round was to mark 4 orange reference markers. In the timed round, there were 8 smaller markers that could not be easily seen without virtual navigation. In both rounds, the approximate locations of the markers were shown in the context of the whole landslip to the user on a separate 2D display (see Figure 40). The task was considered completed when all markers were within 20cm (virtual coordinates) of the solutions.

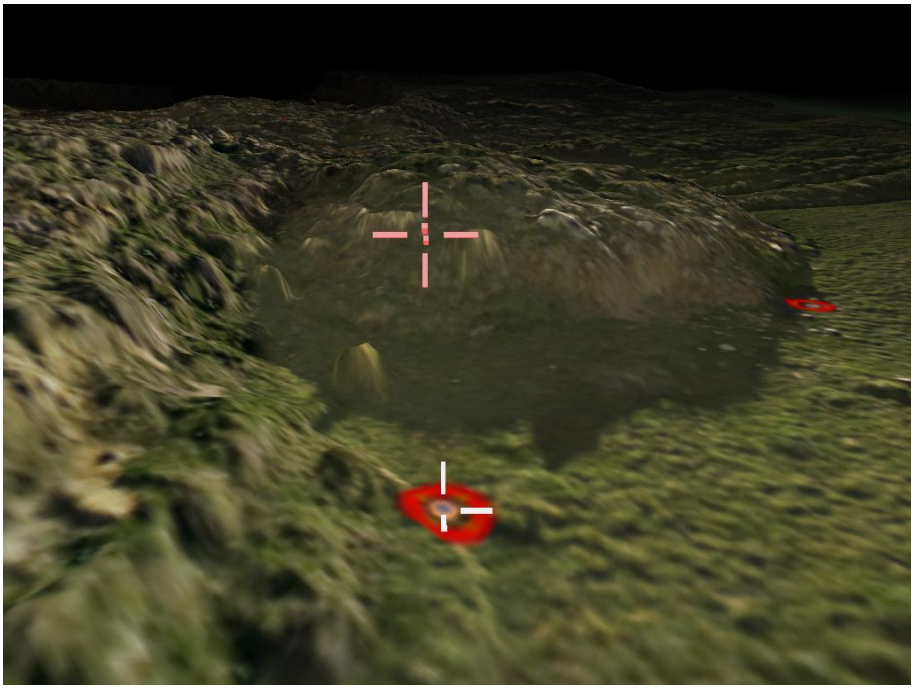

Figure 39: A screenshot from Marking task type, second of the three tasks. The image shows a marker (red ring) being placed on a ground control point; the red and white crosshairs are the true and surface-clamped cursors respectively.

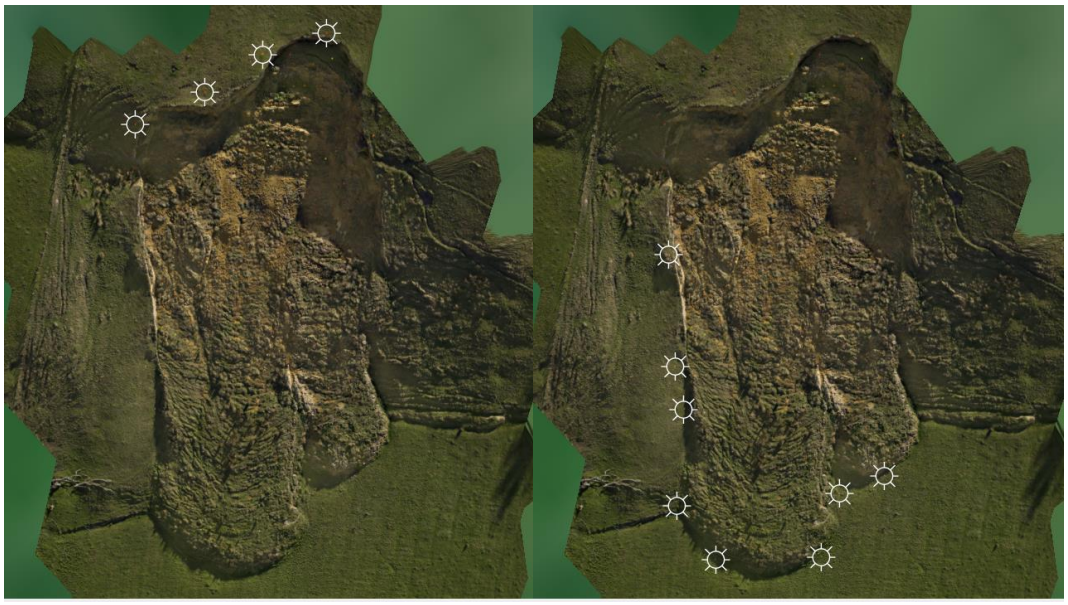

Figure 40: The images used to guide the participants to the marker locations

The third task required the participant to use a vector-based drawing tool to outline certain surface areas, as would be required for digitising terrain areas or highlighting them during a meeting or discussion. The practice task was to roughly outline the entire landslip. The goal task required switching between old and new imagery layers to visualise and outline the area that is newly-eroded, shown in Figure 41. The task was

considered completed when the user's outline did not deviate from the solution at any point by more than 2 virtual metres.

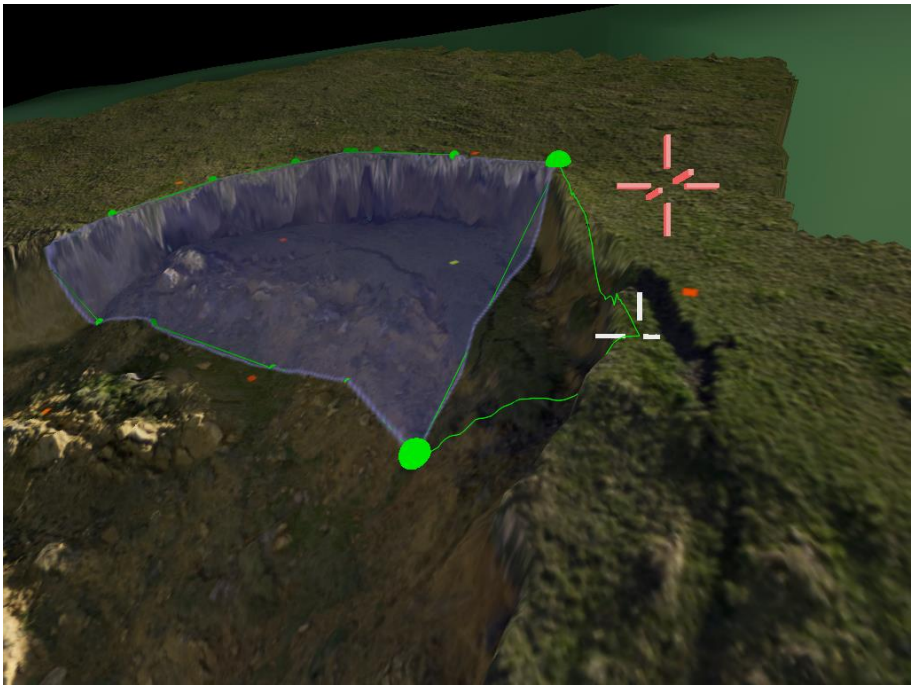

Figure 41: A screenshot of the final Outlining task — The blue highlighted area shows the outline so far. The green lines connecting it to the current clamped cursor position show the new area that would be added if a point were to be added at this point in time.

For the latter two tasks, a cycling approach (as illustrated in Figure 31) was used to switch between the navigation and editing modes. In both cases, a cursor was also used that was constrained to the surface of the terrain model, directly underneath (in cartographic terms) the right virtual hand. On the right hand, *pinch* was used to create new points (to mark the marker locations or define the polygonal outline of the target areas) at the cursor location and *point* was used to delete the closest point to the cursor. On the left hand, *pinch* was used to switch between two imagery layers, which was necessary for the outlining task. Similarly, different buttons were designated to these functions on the 3D mouse.

The experiment was designed as a 2x3 (technique x task) within-subjects counterbalanced test. After completion of the tasks, in addition to being asked open-ended questions about the system and interaction techniques, the participant was asked to rate each technique on a scale of 1-10 for a number of criteria, including appropriateness for the three kinds of task tested (see Figure 42). See Appendix D - User Study II Questionnaire for more details.

4.4 Results

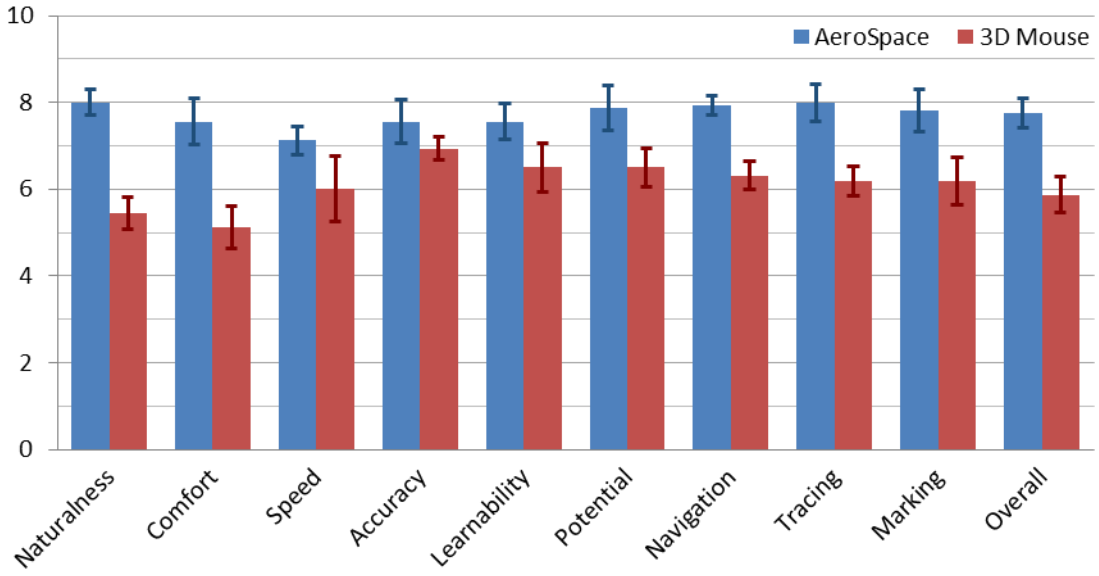

Figure 42: The mean ratings (out of 10) the users gave to the devices across a range of criteria

4.5 Conclusions

This chapter has presented *AeroSpace*, a new elegant technique for 3D spatial input that uniquely allows 7DoF navigation without breaking the principles of direct, simultaneous and bimanual interaction. It has shown that by using a bimanual approach, such navigation can easily be implemented with sub-6DoF tracking.

Chapter 5 - Discussion

I do not fear computers. I fear the lack of them.

— Isaac Asimov

5.1 Introduction

This chapter ties together the work that has been presented in the previous chapters and provides deeper discussion of the results of the two user studies. It also assesses the research opportunities going forward due to the implications of both this research and trends in commercial devices. Based on this analysis it makes a number of suggestions regarding implementation and experiment design for any research that would build on this work.

5.2 Differences in Implementation Capabilities

The first thing to observe about the results of the two major user experiments is the drastic difference between them. While those of the first experiment (as outlined in Chapter 3) were decidedly against the gesture-based approach, those of the second were much more positive. The two systems were implemented in very different ways and it seems possible that, in addition to the fundamental differences in interaction design, these improved results was partly due to the technical superiority of the latter's tracking and classification components in a number of areas. Since the two user studies did not have any participants in common, the effect of these factors merits some speculation. The differences in relation to the mouse and 3D mouse should also be examined.

Spatial accuracy was one such factor; while the ARTrack system in the second gesture system was able to consistently deliver sub-millimetre tracking of the markers, the resolution of the Bumblebee2 used in the first system had a voxel size of around 2.5x2.5x10mm at the typical interaction distance. Furthermore, the realities of computer vision tracking meant that it rarely met this optimal level in practice. While the 3D mouse's coordinates were rate-controlled and the traditional mouse's existed only in logical rather than physical units, making both not directly comparable with the gestural tracking systems, their effective resolution was greater than either of them.

The **pose classification** of the first gesture system was another of its drawbacks. The difference between the target pose states was necessarily large to mitigate issues with classification accuracy, so whereas the second gesture system allowed perfect classification of states that could be moderated with as little as a few millimetres of finger motion, the first system required almost full flexion/extension, adding around half a second of extra delay to each state change. The logical equivalent of poses on the

mice were realised as buttons, which had more or less the same high accuracy and low finger displacement of the glove's poses.

Another factor was **sampling rate**. While the Bumblebee2 camera of the first gesture system had a reasonably smooth capture rate of 48 frames per second, in practice this was more than halved by the limits of the machine on which the system was running, since a number of the image processing steps, especially the stereo correlation phase, were quite computationally expensive. On the other hand, the second system's ARTrack3 was run on a dedicated machine, completely independent of the rest of the system and was capable of providing a consistent 60 frames per second. The sampling rates of the mouse devices were high enough to not be an issue.

From this analysis it is clear that the glove-based gesture system had a number of technical advantages over the first system though it was no better in this regard than the mouse or 3D mouse. While it is not entirely clear how much of the differences between the results of the two gesture systems are due to these implementation factors, it does show that the superiority of the second gesture system over the 3D mouse is not due to technical limitations and must instead be entirely due to the fundamental differences in their interaction design.

5.3 Freehand Interaction

Despite its inferior results, the first gesture system did have one key advantage – the passive nature of the computer vision approach meant that it allowed completely freehand interaction. This requirement was chosen initially in order to reduce strain on and increase the comfort of the user, though the limitations it put on implementation capability evidently made it worse in this regard. However, it is possible that there are many applications and end users for whom encumbered interaction is far from ideal. In terms of geographic virtual environments specifically, this is possibly not a large concession, since expert users could be expected to sit down and use such a device for long enough periods of time to make the inconveniences of the time spent donning and the interruption of everyday hand actions negligible in comparison to the benefits. However, freehand interaction would still be a requirement for short term use cases such as public installations.

The two gesture implementations can be thought of as two data-points on a larger spectrum that encompasses the general trade-off between accurately tracking users' intent and removing encumbrance. While there are many sensor options (such as acoustic, electromyographic and magnetic, among others) for the former, computer vision is currently the only domain that can seriously tackle the latter approach (Rautaray & Agrawal 2012).

Since the Bumblebee2 was chosen as the basis for the first system, the hardware systems available at a commercial level have advanced considerably. In particular there has been a shift toward active devices, which use projected infrared light patterns offset from the camera to detect depth. The advantage of these compared to conventional colour stereo cameras is that they are not disrupted by problems with the lighting of the environment or lack of texture or features in surfaces. The original Kinect device mentioned in previous chapters was the first widespread example of such technology but the limits of its resolution (Khoshelham 2012) and minimum range made it no better than the Bumblebee2 for finger-level detection. More recently, newer Kinect iterations and the much more portable Leap Motion have made such detection possible, though multiple cameras are necessary to avoid occlusion problems (Asteriadis et al. 2013).

However, even with improved detection, classification will still remain as a major hurdle. A major problem is the lack of clear and reliably-detectable definitions of fundamental concepts such as the exact spatial origin of the hand or fingers or the ranges of multiple degrees of freedom within which discrete poses fall. This problem is further exacerbated by the variation in hand, wrist and arm shape that exists between users. Perhaps the largest disadvantage of the freehand approach is the inability to detect finger contact except by estimating the proximity of the finger surfaces in question, which is made difficult to do accurately by the fact that visual occlusion becomes almost unavoidable as the surfaces approach contact. While additional cameras and improved algorithms may make this issue less noticeable than it was in the first system, it is implausible that any freehand system will ever match the performance of worn or held devices in this regard.

5.4 Interaction Principles

The results of the user studies are compatible with the assumptions made about the principles outlined in Section 2.4, though they are inconclusive in some cases.

The qualitative feedback from the first system suggest that a **bimanual** approach seems like a naturally good idea to users. Furthermore it was a defining element of the successful *AeroSpace* approach to navigation and crucial to restricting major input to the positional dimensions, which is important for the implementation and ergonomic reasons mentioned in Section 4.2.2.

Similarly, the user feedback from the first experiment also backed up the assumption on the importance of **simultaneous** interaction and goes a long way to explaining the better results of the gestural system in the second user study, in which there was evidence that translation and rotation were more easily *integrated* under the *AeroSpace* interaction method. While the notion of *simultaneity* is more meaningful in the context of continuous spatial input, it can also be extended to the concurrency of discrete actions. In the case of *AeroSpace*, use of the task-specific actions during bimanual navigation would be impossible without an effect analogous to gaze-based interaction's *Midas Touch* problem (Jacob 1990) on the viewpoint. However, translation with just one hand while the other executed the actions would present no such problems and fits in well with the notion of asymmetric division of labour (Guiard 1987).

Even discounting the possible advantages of simultaneous navigation and manipulation, it is likely that time could be saved by simply eliminating mode-switching, as suggested by one of the participants. Doing so would reduce the user's reliance on either their own memory or feedback from the system for determining the current state. The obvious way to implement such statelessness at a design level would be to have a unique contact/pose for each action, though this could further complicate the hardware and software levels of implementation, especially if *AeroSpace* were ported to a computer-vision approach. Such an alternative would also become impractical if it were applied to tasks requiring a large range of action types. A compromise approach would be to use a modal system with a spatial menu or set of gestures that would make it unnecessary to know the current mode or endure cycling through an inordinate number of modes, at the cost of imperfect simultaneity.

Since neither of the user studies looked specifically at comparing **absolute** and **relative** mappings it is hard to draw any conclusions about them from this work. In some ways the *AeroSpace* system could be considered a relative one, since the mapping between the physical and virtual worlds is modified in cycles of clutching reminiscent of the

pushing and lifting cycles of a mouse, but importantly, the mapping between the physical and virtual hands (i.e., the control-display offset) remained locked. However one possible avenue of future research using a similar system would be to investigate implementing and assessing the impact of making the real-virtual hand offset variable, which would in effect make it a relative system.

The results of the second user study have shown that it is possible to implement a principally **direct** interaction mapping for navigation, though it is not entirely clear whether or not its success is mostly due to the high degree of directness or whether any position-controlled technique could achieve the same results. It is equally possible that gains could be made by removing the hand offset. The most reliable way to do this would be to use a head-mounted display, since it would completely obviate the hand occlusion issue. Recent developments with companies such as Oculus, Samsung, Sony, HTC and Valve aiming to release commercial head-mounted display devices suggests that such fully direct interaction schemes finally have a strong chance at becoming widespread, so the answer to this question may become apparent in the near future.

While the aforementioned design principles that are inspired by the notion of **naturalness** appear to be largely beneficial, it seems that the definition of naturalness itself needs to be re-examined. The physical navigation of the second system highlights this problem; what should have been the most natural axiom of the environment was the least understood and barely utilised. It may be the case that users are paradoxically too accustomed to unnatural interfaces in the context of computer software and underestimate natural capabilities. Perhaps a non-trivial *gulf of confidence* needs to be surmounted before the gains from such techniques can be evident, though there is no reason to suspect that this would outweigh the eventual gains, since it would only be a matter of resolving the relatively small layer of conscious understanding before the bedrock of low-level familiarity could be exploited.

5.5 Methodological Limitations

Though the second user study was a vast improvement over the first in many ways and yielded quite positive results, these results should be interpreted with an understanding of the study's limitations.

One of the important aspects to consider is the degree to which the 3D mouse was a representative device against which to test. Though the ideal approach would be to test against all previous devices and techniques, the sheer number of options make such a goal impractical. The main considerations in choosing the device related to its performance, accessibility and characteristics in terms of the interaction principles outlined in Section 2.4. However, it was similarly impractical to test every permutation of these characteristics, so the level to which each factor was responsible for the difference in results between interaction techniques could only be determined by the qualitative feedback of users. These inferences therefore lack the statistical rigour of the overall comparisons and one potential area for future work would be to test each in isolation.

In many ways, it would also be valuable to test these using variants of the *AeroSpace* gloved approach, so as to remove unwanted variables (such as tracking fidelity and ergonomics) and be able to control the rest separately. The 3D mouse itself would not be capable of such selective comparison; it is inherently limited to being a rate-controlled device that is only bimanual in the sense that it can be held in one hand while the other manipulates its puck.

Bimanuality could be tested by comparing *AeroSpace* to a one-handed approach tracked in 6DoF that mapped rotation directly to hand orientation, as mentioned in Section 4.2.2. It is not immediately clear how directness would be preserved for the scalar degree of freedom, though mapping it to the flexion of a specific digit may be possible. Such a design change would likely sacrifice much in terms of implementation simplicity in any case. The naïve way to reduce **simultaneity** could be implemented by merely adding to the software a heuristic similar to that described in Section 3.4.2.6 to decide which degrees of freedom to control. Explicit modes activated by discrete actions might be more reliable, but might also require extra finger contact points on the gloves. **Directness** would be easier to remove; the existing *AeroSpace* system could be altered to use the same mapping but constantly reapply some fraction of the transformation while contacts are held down. It could also be implemented on one hand, by using the 3DoF position relative to the start of the transformation action to control the three rotation axes, which would necessitate separate contacts for non-simultaneous translation, rotation and scaling modes, giving a good baseline implementation not meeting any of the target criteria.

5.6 GIS Implications

The results of the second user study have shown that a gestural interaction method is effective for not only generic spatial navigation, but is also superior for other geovisualisation tasks. While it might not be possible to draw conclusions from these results that would be applicable to more traditional GIS interaction, they do suggest that such natural interaction is the ideal spatial interaction method for 3D geovisualisations.

Although it was not one of the interaction capabilities that was implemented in software and tested, pointing-based interactions are perfectly suited to the *AeroSpace* method, since a ray can be projected from the index finger during the point pose. This could be used for pointing and tracing tasks and would be similar to the way deictic gestures work in real word human-to-human interaction. There are also a number of ways that a gestural system incorporating *AeroSpace* could be designed to include traditional GIS capabilities. The simplest would be to place the traditional 2D GUI in the 3D interaction space and allow a pointing gesture to take the place of the mouse. A further clutch or toggle pose might be necessary to hide the GUI which would occlude large sections of the virtual environment when visible. A 3D input system also allows the spatial menu analogy to be scaled up to the higher dimension, by creating 3D volumes rather than 2D surfaces wherein choices are selected by placing the hand or index finger within a menu item instead of pointing at it. Another option would be to abandon spatial menus altogether. Using discrete hand gestures in the traditional sense of a unique hand shape or path would be one way of replacing this functionality. One major requirement that remains unaddressed is textual input, which is necessary for many GIS operations. Speech input is one way, but it is prone to error and unlikely to deal well with the arbitrary range of possible layer and location names. It remains to be seen if the gestural system and its efficiency presented in Chapter 4 could be extended to allow such functionality.

Another area that warrants investigation is collaboration. A screen-based approach as presented in Chapter 4 would present problems for local collaboration, since rendering according to a head-tracked perspective only works for one user. However, using head-mounted displays or collaborating remotely would still be possible. This would be particularly useful for emergency management, where officials in disparate locations may need to convene a teleconference and discuss geographic details at short notice.

Since the basic requirements of *AeroSpace* include the position and at least direction of the user's head and hands, such a collaborative implementation would also make it possible to see where colleagues are facing and pointing. It is currently unclear how the mapping between physical and virtual spaces would work in such a situation. One approach would be to have a separate virtual cameras for each user. This might hamper collaboration, since it would allow them to be at vastly different locations and even scales within the virtual world. The alternative would be for them to share a single set of physical-to-virtual transformation parameters; i.e., any given point in the real physical space of the room one user inhabits would consistently map to a single point in each of his/her collaborator's rooms, regardless of virtual navigation. This would mean that ever user's virtual viewpoint would change as any one performed virtual navigation. It is also not immediately obvious what should happen when more than one user attempted virtual navigation at the same time. Mathematically it could be easily realised by effecting the compound transformation of the two individual transformations, though it would have to be calculated and presented incrementally since such series of transformations could not be guaranteed to be commutative.

Changing to an incremental transformation model would also enable dynamic physics effects similar to those in some touch-based map applications, allowing the world to maintain velocity on clutch release, continuing to move until brought to a stop by friction or another clutch event. Releasing two-handed actions would give angular and scalar velocity; it would also do so independently of translational velocity if only one of the two hands were released, allowing one to, for example, set the globe spinning with one hand while it stays fixed to the other's position.

5.7 Further Implications

This work has also made contributions with implications that go beyond the specific scope of geovisualisation.

The novel approach (detailed in Section 3.4.2) of using forearm orientation with a cross-section hand-segmentation heuristic to normalise the starting point of DTW template-matching has implications generally for applications classifying hand pose.

Perhaps more importantly, the *AeroSpace* approach to navigation is applicable to a broad range of 3D spatial applications. Since its approach to 7DoF navigation can be

thought of as moving the world around the user rather than moving the user through the world, it could be applied to both navigation and the manipulation of objects. Which of the two to affect could be decided by the current selection state or an extra contact mapped to toggling between world and object modes. Another possibility would be to decide based on which hand initiated the action. This would destroy *AeroSpace*'s symmetry, but in any case asymmetry would possibly be desirable to deal with the manipulation of objects with fixed scale; the two-point principle of consistent direct mapping mentioned in Section 4.2.2 would be relaxed for one hand such that it would only affect orientation relative to the other hand and not scale.

The other general contributions of *AeroSpace* relate to implementation complexity. It illustrates that 6DoF tracking is not necessary for natural 6DoF interaction and can be replaced with simpler and more robust 3DoF and 5DoF tracking in similar applications. Just as importantly, it shows that core interactions are achievable with finger contacts instead of the full pose information offered by most existing data gloves, offering a design that is both reliable and significantly cheaper to implement.

Chapter 6 - Conclusion

It doesn't stop being magic just because you know how it works.

— Terry Pratchett, *The Wee Free Men*

The primary aim of this thesis was to show that natural interaction principles can be used to develop a gestural approach that improves interaction with geovisualisations. It also aimed to give insights into which design principles are beneficial to such interaction and the best manner in which to implement it.

The first user study failed to address the primary question but yielded a few key results. Chiefly, it gave qualitative evidence that ideal gestural interaction would be simultaneous and bimanual, but also identified technical issues with the chosen computer vision implementation approach. Additionally, the novel approach of using arm orientation to optimise hand pose classification with DTW proved useful and is a ripe area for further investigation.

The results of the second user study clearly showed that a gestural approach can outperform a leading commercial device specifically designed for 3D interaction in the key areas of comfort and efficiency for geovisualisation tasks incorporating navigation. These results also underline the importance of making interaction bimanual, simultaneous and at least position-controlled if not properly direct. The next step for future research should be to test a fully-direct version of *AeroSpace* in a modern HMD VR environment.

This work has also given reasoned arguments for the importance of the largely ignored concepts that are 7DoF and physical navigation while also showing them to be, contrary to expectations, sufficiently unfamiliar to ordinary users to mean that most would need some non-trivial level of training or practice to make full use of them. The design of efficient such familiarisation techniques may be essential to ensuring the imminent resurgence of VR captures as broad a base of users as possible.

In terms of technical contributions, this thesis has also presented a unique combination of image-processing techniques for hand pose detection as well as a model of glove-based interaction suitable for general spatial interaction in VR that nonetheless reduces tracking complexity and ergonomic strain.

It seems assured that both geovisualisation and natural interaction will witness rapid growth over the coming years and it is the hope of the author that this work can go some way to guiding subsequent work in this area.

Bibliography

Accot, J. & Zhai, S., 2003. Refining Fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '03*. ACM Press, pp. 193–200.

Asteriadis, S. et al., 2013. Estimating human motion from multiple Kinect sensors. In *Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications - MIRAGE '13*. ACM, pp. 3–8.

Balakrishnan, R. & Kurtenbach, G., 1999. Exploring Bimanual Camera Control and Object Manipulation in 3D Graphics Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '99*. ACM Press, pp. 56–62.

Ball, R., North, C. & Bowman, D., 2007. Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '07*. ACM Press, pp. 191–200.

Bartoschek, T. et al., 2014. Gestural Interaction with Spatiotemporal Linked Open Data. *OSGeo Journal*, 13(1), pp.60–67.

Bi, X., Chelba, C. & Ouyang, T., 2012. Bimanual gesture keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology - UIST '12*. ACM, pp. 137–146.

Bierwirth, D., 2008. *3D Interaction Widget: A Metaphor for 2D and 3D Lens Interaction*. Hasso-Plattner-Institute.

Bowman, D.A., 2013. 3D User Interfaces. In *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* The Interaction Design Foundation.

Bowman, D.A. et al., 2004. *3D User Interfaces: Theory and Practice*, Addison Wesley Longman Publishing.

Bowyer, K.W., Chang, K. & Flynn, P., 2006. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Computer Vision and Image Understanding*, 101(1), pp.1–15.

Capobianco, A., Veit, M. & Bechmann, D., 2009. A Preliminary Study of Two-Handed Manipulation for Spatial Input Tasks in a 3D Modeling Application. In *Computer-Aided Design of User Interfaces VI*. London: Springer, pp. 77–88.

Crampton, J.W., 2002. Interactivity types in geographic visualization. *Cartography and Geographic Information Science*, 29(2), pp.85–98.

Cutler, L.D., Frohlich, B. & Hanrahan, P., 1997. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics - SI3D '97*. ACM, pp. 107–114.

Cutting, J., 1997. How the eye measures reality and virtual reality. *Behavior Research Methods, Instruments, & Computers*, 29(1), pp.27–36.

Daiber, F., Schöning, J. & Krüger, A., 2009. Whole body interaction with geospatial data. In *Lecture Notes in Computer Science*. pp. 81–92.

Darken, R.P. & Durost, R., 2005. Mixed-dimension interaction in virtual environments. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology - VRST '05*. ACM Press, pp. 38–45.

Ekberg, F., 2007. *An approach for representing complex 3D objects in GIS applied to 3D properties*. University of Gävle.

Elmezain, M. et al., 2009. A Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition. *International Journal of Electrical, Computer, and Systems Engineering*, 3(3), pp.156–163.

Elmqvist, N. & Fekete, J.-D., 2008. Semantic Pointing for Object Picking in Complex 3D Environments. In *Proceedings of Graphics Interface - GI '08*. ACM, pp. 243–250.

El-Shimy, D., Marentakis, G. & Cooperstock, J.R., 2009. Tech-note: Multimodal feedback in 3D target acquisition. In *2009 IEEE Symposium on 3D User Interfaces*. IEEE, pp. 95–98.

Fitts, P.M., 1954. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, 47(6), pp.381–391.

Frampton, A., Solomon, J.D. & Wong, C., 2012. *Cities Without Ground: A Hong Kong Guidebook*, ORO Editions.

Garner, W.R. & Felfoldy, G.L., 1970. Integrality of stimulus dimensions in various types of information processing. *Cognitive Psychology*, 1(3), pp.225–241.

Garstka, J. & Peters, G., 2011. View-dependent 3D projection using depth-image-based head tracking. In *8th IEEE International Workshop on Projector-Camera Systems - PROCAMS '11*. pp. 52–59.

Grandjean, E., 1982. *Fitting the task to the man: an ergonomic approach*, London: Taylor & Francis.

Guiard, Y., 1987. Asymmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *Journal of Motor Behavior*, 19(4), pp.486–517.

Hancock, M.S. et al., 2006. Rotation and Translation Mechanisms for Tabletop Interaction. In *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems - TABLETOP '06*. IEEE, pp. 79–88.

Hassanpour, R., Wong, S. & Shahbahrami, A., 2008. Vision-Based Hand Gesture Recognition for Human Computer Interaction: A Review. In *IADIS International Conference on Interfaces and Human Computer Interaction 2008 - IHCI 2008*. IADIS, p. 125.

Hilliges, O. et al., 2012. HoloDesk. In *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, pp. 2421–2430.

Huang, T.S., 2000. Modeling the constraints of human hand motion. In *Proceedings of the Workshop on Human Motion - HUMO '00*. IEEE Computer Society, pp. 121–126.

Hwang, M.-H. et al., 2013. Spatiotemporal transformation of social media geostreams. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on GeoStreaming - IWGS '13*. ACM Press, pp. 12–21.

InfoStrat, 2011. Bing Maps controlled with Kinect 3D-sensing Technology. Available at: www.youtube.com/watch?v=G2llpgpOV5Q.

Jacob, R.J.K. et al., 1994. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction - TOCHI*, 1(1), pp.3–26.

Jacob, R.J.K., 1990. What you look at is what you get: eye movement-based interaction techniques. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems - CHI '90*. ACM Press, pp. 11–18.

Jiang, B., Huang, B. & Vasek, V., 2003. Geovisualization for planning support systems. In S. Geertman & J. Stillwell, eds. *Planning Support Systems in Practice*. Springer: Berlin, pp. 177–191.

Kamel Boulos, M.N. et al., 2011. Web GIS in practice X: a Microsoft Kinect natural user interface for Google Earth navigation. *International Journal of Health Geographics*, 10(1).

de Kemp, E. a. et al., 2011. 3D GIS as a support for mineral discovery. *Geochemistry: Exploration, Environment, Analysis*, 11(2), pp.117–128.

Kersting, O. & Döllner, J., 2002. Interactive 3D visualization of vector data in GIS. In *Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems - GIS '02*. ACM Press, pp. 107–112.

Keskin, C., Erkan, A. & Akarun, L., 2003. Real Time Hand. Tracking and 3D Gesture Recognition for Interactive. Interfaces Using HMM. In *Proceedings of the Joint International Conference on Artificial Neural Networks and International Conference on Neural Information Processing - ICANN/ICONIP 2003*. Springer-Verlag.

Khoshelham, K., 2012. Accuracy Analysis of Kinect Depth Data. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(5), pp.133–138.

Knoedel, S. & Hachet, M., 2011. Multi-touch RST in 2D and 3D spaces: Studying the impact of directness on user performance. In *Proceedings of the IEEE Symposium on 3D User Interfaces - 3DUI '11*. IEEE, pp. 75–78.

Knust, C. & Buchroithner, M.F., 2014. Principles and Terminology of True-3D Geovisualisation. *The Cartographic Journal*, 51(3), pp.191–202.

Kulik, A., 2009. Building on realism and magic for designing 3D interaction techniques. *IEEE Computer Graphics and Applications*, 29(6), pp.22–33.

Kulik, A. et al., 2009. The influence of input device characteristics on spatial perception in desktop-based 3D applications. In *Proceedings of the IEEE Symposium on 3D User Interfaces - 3DUI '09*. IEEE, pp. 59–66.

Kurakula, V., 2007. *A GIS-Based Approach for 3D Noise Modelling Using 3D City Models*. International Institute for Geo-Information Science and Earth Observation.

Kwan, M. & Lee, J., 2003. Geovisualization of Human Activity Patterns Using 3D GIS : A Time-Geographic Approach. In *Spatially Integrated Social Science: Examples in Best Practice*. Oxford University Press.

Kwan, M.-P. & Lee, J., 2005. Emergency response after 9/11: the potential of real-time 3D GIS for quick emergency response in micro-spatial environments. *Computers, Environment and Urban Systems*, 29(2), pp.93–113.

LaViola Jr., J., 1999. *A survey of Hand Posture and Gesture Recognition Techniques and Technology*. Brown University.

Lee, J., 2007. A Three-Dimensional Navigable Data Model to Support Emergency Response in Microspatial Built-Environments. *Annals of the Association of American Geographers*, 97(3), pp.512–529.

Lee, J. & Zlatanova, S., 2008. A 3D data model and topological analyses for emergency response in urban areas. In *Geospatial Information Technology for Emergency Response*. CRC Press, pp. 143–167.

Lee, M., Green, R. & Billinghurst, M., 2008. 3D Natural Hand Interaction for AR Applications. In *Proceedings of the 23rd International Conference on Image and Vision Computing New Zealand - IVCNZ 2008*. IEEE.

Lemmerman, D. & Laviola, J., 2007. An Exploration of Interaction-Display Offset in Surround Screen Virtual Environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces - 3DUI '07*. IEEE, pp. 9–15.

Lucieer, A., Jong, S.M.D. & Turner, D., 2014. Mapping landslide displacements using Structure from Motion (SfM) and image correlation of multi-temporal UAV photography. *Progress in Physical Geography*, 38(1), pp.97–116.

MacEachren, A.M. & Kraak, M.-J., 2001. Research Challenges in Geovisualization. *Cartography and Geographic Information Science*, 28(1), pp.3–12.

MacEachren, A.M. & Taylor, D.R.F., 1994. *Visualization in modern cartography*, Pergamon.

MacKenzie, I.S., Sellen, A. & Buxton, W., 1991. A comparison of input devices in element pointing and dragging tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '91*. ACM Press, pp. 161–166.

Maguire, D., 1991. An overview and definition of GIS. *Geographical Information Systems: Principles and Applications*, 1, pp.9–20.

Malerczyk, C. & Engelke, T., 2009. Intuitive Interaction with VR Applications Using Video-based Gesture Recognition. In *Proceedings of the 2nd Workshop on Software Engineering and Architectures for Realtime Interactive Systems - SEARIS @ VR 2009*. IEEE, pp. 37–40.

Martinet, A., Casiez, G. & Grisoni, L., 2010. The effect of DOF separation in 3D manipulation tasks with multi-touch displays. In *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology - VRST '10*. ACM Press, pp. 111–118.

Masliah, M.R. & Milgram, P., 2000. Measuring the allocation of control in a 6 degree-of-freedom docking experiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '00*. ACM Press, pp. 25–32.

May, R., 2004. *Toward Directly Mediated Interaction in Computer Supported Environments*. University of Washington.

Mine, M.R., Brooks, Frederick P, J. & Sequin, C.H., 1997. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *24th International ACM Conference on Computer Graphics & Interactive Techniques - SIGGRAPH '97*. ACM Press/Addison-Wesley Publishing Co., pp. 19–26.

Mitra, S. & Acharya, T., 2007. Gesture Recognition: A Survey. *IEEE Transactions On Systems, Man, And Cybernetics, Part C: Applications And Reviews*, 37(3), pp.311–324.

Moehring, M. & Froehlich, B., 2011. Effective manipulation of virtual objects within arm's reach. In *Proceedings of the 2011 IEEE Virtual Reality Conference - VR '11*. IEEE, pp. 131–138.

Nielsen, A., 2007. *A Qualification of 3D Geovisualisation*. Aalborg University.

Nielsen, M. et al., 2004. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. *Gesture-Based Communication in Human-Computer Interaction*, pp.409–420.

Nielsen, M. et al., 2003. *A procedure for developing intuitive and ergonomic gesture interfaces for Man-Machine interaction*. Aalborg University.

O'Hagan, R.G., Zelinsky, A. & Rougeaux, S., 2002. Visual Gesture Interfaces for Virtual Environments. *Interacting with Computers*, 14(3), pp.231–250.

Oakley, I. & O'Modhrain, S., 2005. Tilt to Scroll: Evaluating a Motion Based Vibrotactile Mobile Interface. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. IEEE, pp. 40–49.

Owen, R. et al., 2005. When it gets more difficult, use both hands: exploring bimanual curve manipulation. In *Proceedings of Graphics Interface 2005 - GI '05*. pp. 17–24.

Özacar, K., Takashima, K. & Kitamura, Y., 2013. Direct 3D object manipulation on a collaborative stereoscopic display. In *Proceedings of the 1st Symposium on Spatial User Interaction - SUI '13*. ACM Press, pp. 69–72.

Point Grey, 2010a. Point Grey stereo vision product catalogue. Available at: http://www.ptgrey.com/products/Point_Grey_stereo_catalog.pdf.

Point Grey, 2010b. Triclops Product Datasheet. Available at: http://www.ptgrey.com/products/triclopsSDK/triclops.pdf.

Point Grey, 2010c. Triclops SDK samples gallery. Available at: http://www.ptgrey.com/products/triclopsSDK/samples.asp.

Price, P., 1934. The geology and ore deposits of the Horne mine, Noranda, Quebec. *Transactions Of The Canadian Institute Of Mining And Metallurgy*, 37, pp.108–140.

Ragan, E.D. et al., 2013. Studying the effects of stereo, head tracking, and field of regard on a small-scale spatial judgment task. *IEEE Transactions on Visualization and Computer Graphics*, 19(5), pp.886–896.

Raja, Y., McKenna, S. & Gong, S., 1997. Segmentation and tracking using colour mixture models. In *Proceedings of the 3rd Asian Conference on Computer Vision - ACCV '98*. Springer, pp. 607–614.

Rautaray, S.S. & Agrawal, A., 2012. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, 43(1), pp.1–54.

Rizzo, A.A. et al., 2005. Development of a benchmarking scenario for testing 3D user interface devices and interaction methods. In *Proceedings of the 11th International Conference on Human Computer Interaction - HCII 2005*. Lawrence Erlbaum Associates.

Robles De La Torre, G., 2006. The Importance of the Sense of Touch in Virtual and Real Environments. *IEEE Multimedia*, 13(3), pp.24–30.

Roth, R.E., 2013. Interactive maps: What we know and what we need to know. *Journal of Spatial Information Science*, 6(6), pp.59–115.

Santosh, K., 2010. Use of dynamic time warping for object shape classification through signature. *Kathmandu University Journal of Science, Engineering and Techonology*, 6(1), pp.33–49.

Schlattmann, M., Broekelschen, J. & Klein, R., 2009. Real-Time Bare-Hands-Tracking for 3D Games. In *Proceedings of the IADIS International Conference Game and Entertainment Technologies - GET '09*. IADIS Press, pp. 59–66.

Schlattmann, M. & Klein, R., 2009. Efficient bimanual symmetric 3D manipulation for markerless hand-tracking. In *Proceedings of the Virtual Reality International Conference - VRIC '09*.

Schultheis, U. et al., 2012. Comparison of a two-handed interface to a wand interface and a mouse interface for fundamental 3D tasks. In *Proceedings of the IEEE Symposium on 3D User Interfaces - 3DUI '12*. IEEE, pp. 117–124.

Shao, Y. et al., 2011. 3D Geological Modeling and Its Application under Complex Geological Conditions. *Procedia Engineering*, 12, pp.41–46.

Shepherd, I.D.H. & Bleasdale-shepherd, L.D., 2008. Towards effective Interaction in 3D data visualizations: what can we learn from videogames technology. In *Proceedings of the International Conference on Virtual Geographic Worlds*.

Shin, M.C., Tsap, L. V & Goldgof, D.B., 2003. Towards Perceptual Interface for Visualization Navigation of Large Data Sets. In *Proceedings of Computer Vision and Pattern Recognition Workshop - CVPRW '03*. IEEE Computer Society, pp. 48–53.

Shneiderman, B., 1983. Direct Manipulation: A Step Beyond Programming Languages. *Computer*, 16(8), pp.57–69.

Shrirao, N.A., Reddy, N.P. & Kosuri, D.R., 2009. Neural network committees for finger joint angle estimation from surface EMG signals. *BioMedical Engineering OnLine*, 8(2).

Slocum, T.A. et al., 2001. Cognitive and Usability Issues in Geovisualization. *Cartography and Geographic Information Science*, 28(1), pp.61–75.

Song, L. & Takatsuka, M., 2005. Real-time 3D Finger Pointing for an Augmented Desk. In *Proceedings of the 6th Australasian User Interface Conference - AUIC '05*. ACM, pp. 99–108.

Song, P. et al., 2012. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, pp. 1297–1306.

Swan, J.E. et al., 2003. A comparative study of user performance in a map-based virtual environment. In *Proceedings of the 2003 IEEE Virtual Reality Conference - VR '03*. IEEE Computer Society, pp. 259–266.

Sweetser, J., Grunnet-Jepsen, A. & Panchanathan, G., 2008. Absolute pointing and tracking based remote control for interactive user experience. In *Proceedings of the 1st International Conference on Designing Interactive User Experiences for TV and Video - uxTV' 08*. ACM Press, pp. 155–164.

Technology Assessment Group, 2008. The Economic Payback of 3D Mice for CAD Design Engineers. *Technology Assessment Group*. Available at: http://www.3dconnexion.com/fileadmin/user_upload/manuals_docs/english_intl/3dx_whitepaper_cadpayback_en_intl.pdf.

Ulinski, A.C. et al., 2009. Selection performance based on classes of bimanual actions. In *Proceedings of the 2009 IEEE Symposium on 3D User Interfaces - 3DUI '09*. IEEE, pp. 51–58.

Vafaei, F., 2013. *Taxonomy of Gestures in Human Computer Interaction*. North Dakota State University of Agriculture and Applied Science.

VanHorn, J.E. & Mosurinjohn, N. a., 2010. Urban 3D GIS Modeling of Terrorism Sniper Hazards. *Social Science Computer Review*, 28(4), pp.482–496.

Vatavu, R.D., Pentiuc, Ş. & Chaillou, C., 2005. On Natural Gestures for Interacting with Virtual Environments. *Advances in Electrical and Computer Engineering*, 24(5).

Wachs, J. et al., 2006. A Real-Time Hand Gesture Interface for a Medical Image Guided System. In *Proceedings of the Ninth Israeli Symposium on Computer-Aided Surgery, Medical Robotics, and Medical Imaging - ISRACAS 2006*. pp. 175–185.

Wang, G. et al., 2011. Mineral potential targeting and resource assessment based on 3D geological modeling in Luanchuan region, China. *Computers & Geosciences*, 37(12), pp.1976–1988.

Wang, R., Paris, S. & Popović, J., 2011. 6D Hands : Markerless Hand Tracking for Computer Aided Design. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology - UIST '11*. ACM, pp. 549–557.

Wartell, Z., Ribarsky, W. & Hodges, L., 1999. Third-person navigation of whole-planet terrain in a head-tracked stereoscopic environment. In *Proceedings of the 1999 IEEE Virtual Reality Conference - VR '99*. IEEE Computer Society, pp. 141–148.

Wigdor, D. & Wixon, D., 2011. *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*, Elsevier.

Wisniewski, P.K. et al., 2009. Grounding Geovisualization Interface Design: A Study of Interactive Map Use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '09*. ACM Press, pp. 3757–3762.

Wolff, M. & Asche, H., 2010. Towards 3D Tactical Intelligence Assessments for Crime Scene Analysis. In *Proceedings of the 10th International Conference on Computational Science & its Applications - ICCSA 2010*. Springer, pp. 346–360.

Xu, Z. et al., 2009. Hand Gesture Recognition and Virtual Game Control Based on 3D Accelerometer and EMG Sensors. In *Proceedings of the 14th International Conference on Intelligent User Interfaces - IUI '09*. ACM, pp. 401–405.

Zeleznik, R.C., Forsberg, A.S. & Strauss, P.S., 1997. Two pointer input for 3D interaction. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics - SI3D '97*. ACM Press, pp. 115–120.

Zhai, S., 1995. *Human performance in six degree of freedom input control*. University of Toronto.

Zhai, S. & Milgram, P., 1998. Quantifying coordination in multiple DOF movement and its application to evaluating 6 DOF input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '98*. ACM Press, pp. 320–327.

Zhou, K. & Guo, M., 2011. Virtual reality simulation system for underground mining project. In *Virtual Reality*. InTech, pp. 615–633.

Appendices

Appendix A - Bumblebee2 Benchmarking Report

After doing some initial testing, I was concerned that the results of hand segmentation using depth data captured and processed by the Bumblebee2 and its associated Triclops software were sub-optimal due to some issue with my setup. This appendix discusses such possible issues and presents images and data taken using my setup as well as those from the camera's manufacturer - Point Grey, with the aim of validating the former by reproducing their benchmarks.

It was anticipated that any differences in results would be traceable back to differences in either the software or hardware. In the case of software, this would be any number of parameters used during processing. The main potential issue with hardware would have been error in calibration.

A.1 Basic Images

Shown below are two sets of images from the Point Grey website. They illustrate examples of how successful the stereo system is with large objects at varying distances. For both, I tried to capture images with similar properties to see if I can achieve similar results.

The first example shows a well-textured scene with identifiable objects across a range of distances. I created an image with large flat surfaces that were similarly not perpendicular to the camera's line-of-sight. Also, I tried to place objects at varying distances from the camera.

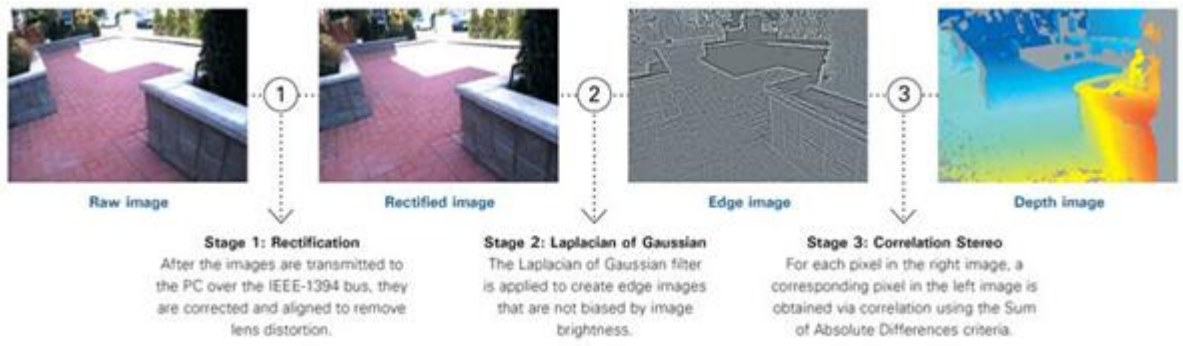

Figure 44: Bumblebee2's image process as advertised by Point Grey (2010a)

Figure 45: Comparable depth test situation featuring objects at various distances

In terms of accuracy, the results were similar; there is a smooth progression in the colours between close and distant objects. The only noticeable exceptions to this are the visual artefacts on the right (corresponding to outside the boundaries of one of the cameras at the true distance) and the keys of the keyboard. However, the results vary obviously in that Point Grey's depth image has valid values for most of the image, whereas my image only shows the depth of the obvious objects, and often only part of their outlines. The reason for this is that, whereas Point Grey's image is well-textured, the majority of my image is not; the different parts of the surface of the walls and desk are not distinct enough from each other for correlation to be meaningful. The only combinations of parameters that would give those areas depth values would cause those values to be significantly erroneous.

The second reference image (Figure 46, below), shows two people at different distances, with one partially obscured by the other. The depth image seems to have been produced using a relatively large stereo (and/or edge) mask, as the details and outlines of the two people do not show up clearly. I tried to recreate a similar situation by taking a shot of two people positioned in roughly the same way.

Figure 46: Further samples from Point Grey (Point Grey 2010c), of figures at closer ranges

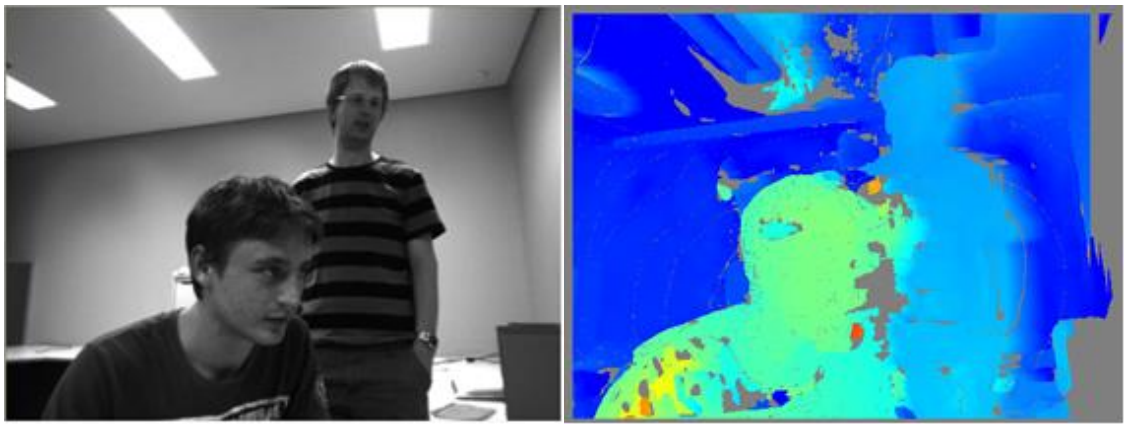

Figure 47: Comparable test image of two people at different distances with stereo result

The results, shown in Figure 47, were largely the same; the only major difference being an increased unmatched area where the two people overlap. This is probably due to a combination of my setup being at a closer range (thus creating a larger region obscured in one side of the original frame) and a difference in parameters (either mask size or validation).

A.2 Hand Segmentation

The following set of images from Point Grey show a user's hand being detected on the basis of the distance of its pixels from the camera. The pixels determined to be in the foreground mostly correspond to the hand's pixels; however an undefined segmentation step was used to generate the final mask that defines the clear outline of the hand. Also, the hand in this image was not close to any other objects that might have significantly confused the stereo matching process.

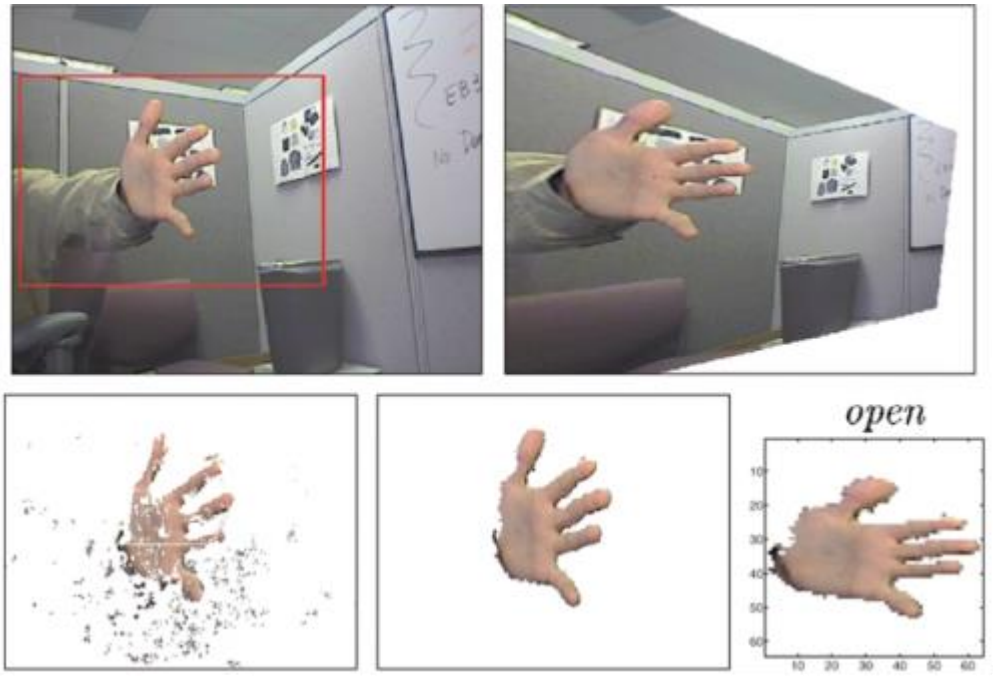

Figure 48: Example depth-based segmentation from the Triclops product datasheet (Point Grey 2010b)

Figure 49, below, shows a test set of hand images that I created to roughly imitate those of Point Grey, along with the depth images from what seemed to be the most appropriate set of parameters.

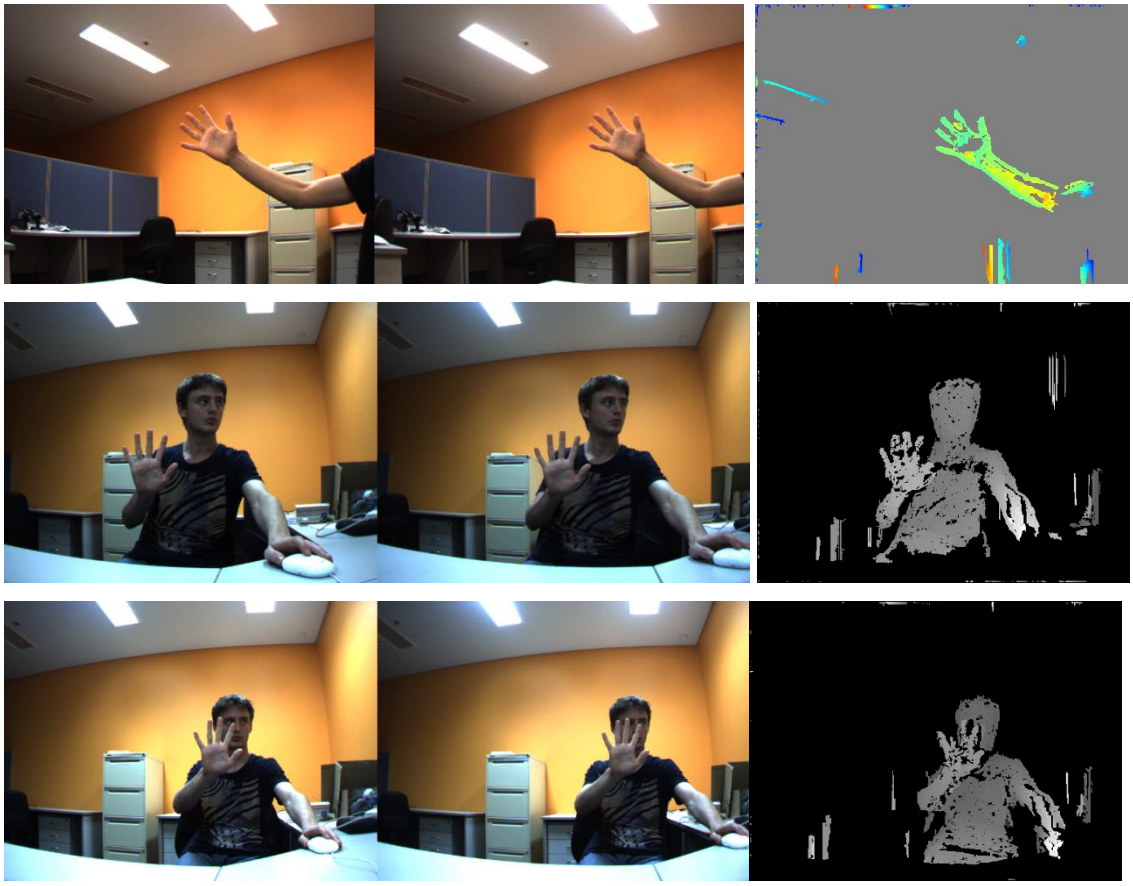

Figure 49: Three sets of hand images with differing levels of obfuscation — a hand in front of a simple background; a hand in front of body, and; a hand in front of face

The first and second set of images appeared to have at least as good a level of detail as Point Grey's above test result (before post-processing). I deliberately chose to test a more difficult potential situation for the third set of images: the hand obscuring the user's face. The similarities in colouring and high detail of the face seemed to negatively affect the ability of the stereo algorithm to pick up the shape of the hand.

Without implementing the subsequent step of segmenting the hand based on this depth data, it was not clear how readily a contour of the hand could be produced from my images. In any case, it was clear that there were sufficient valid pixels registered at sufficiently accurate depths, to at least track the hand's position in three dimensions (under the reasonable assumption that the hand would always be the closest point to the camera).

A.3 Calibration

Finally, I did a limited test of the Bumblebee2's calibration. This was important, since even small issues with either the lenses or their physical placement could have severely

hampered the results of the stereo correlation process. Figure 50, below, shows two sections (100 x 76 pixels) of the same frame from both the left and right cameras. The overlaid white lines (1-pixel thick) are at the same vertical position and the notebook pictured was perpendicular to both the direction of the camera and the ground plane on which it rested.

Figure 50: Calibration test images from the two halves of a frame

This simple test showed the vertical calibration error to be less than a pixel. The same was true of other sections of the frame that I tested. Although this test does not prove that the calibration is perfect, it does prove that the two rectified images are vertically aligned, which means that though there is a small chance the depth values may be skewed, the chance that the two occurrences of a given object will be matched should not be decreased.

A.4 Stereo Precision

In order to test the precision of the stereo results, I set up an experiment using images of an object at different distances from the camera. Accuracy was not as much of a concern, since the position of the camera relative to the user's hands was considered irrelevant to interaction, whereas imprecision could result in unacceptable jerkiness. The base image pair I used for testing was of the object at approximately 1.45m from the camera, in the centre of its field of view. More images were taken at further distances (10, 12, 15, 20, 30 and 40mm, relative to the first image pair).

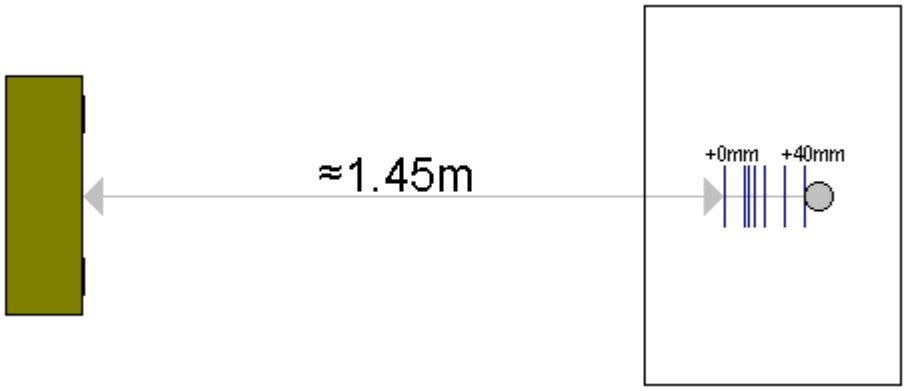

Figure 51: Depth precision test setup with the test object at the relative position of +40mm

I ran the stereo software on each pair of images and obtained the results for the image pixels of the object (using the same pixel locations in all images). The same stereo configuration was used for all images, with a disparity range of 21 to 42 pixels (normalised by the stereo software up to 255, which are the units used in the following results). Assuming successful stereo matching, an increase in distance should have caused a decrease in these values.

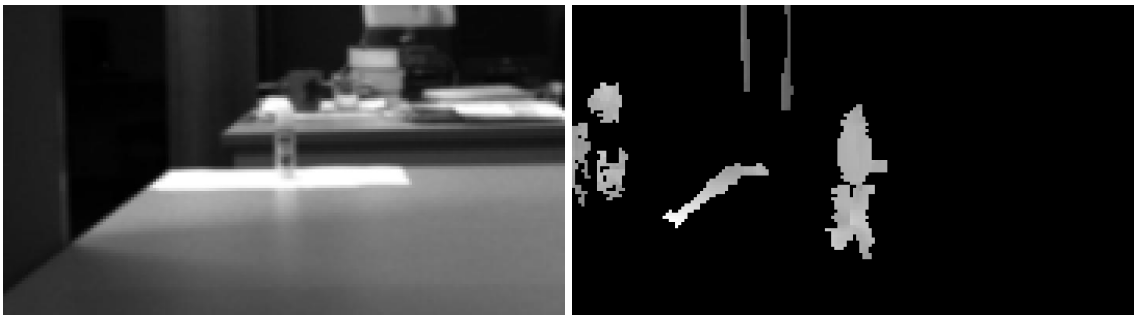

Figure 52: Subregions of one side of the input image pair and the resulting depth image

The figures I recorded were the mean values of a region of pixels, with size varying from 1x1 to 7x7 pixels in size. Due to the small size and cylindrical shape of the object, the larger regions recorded slightly further mean distances.

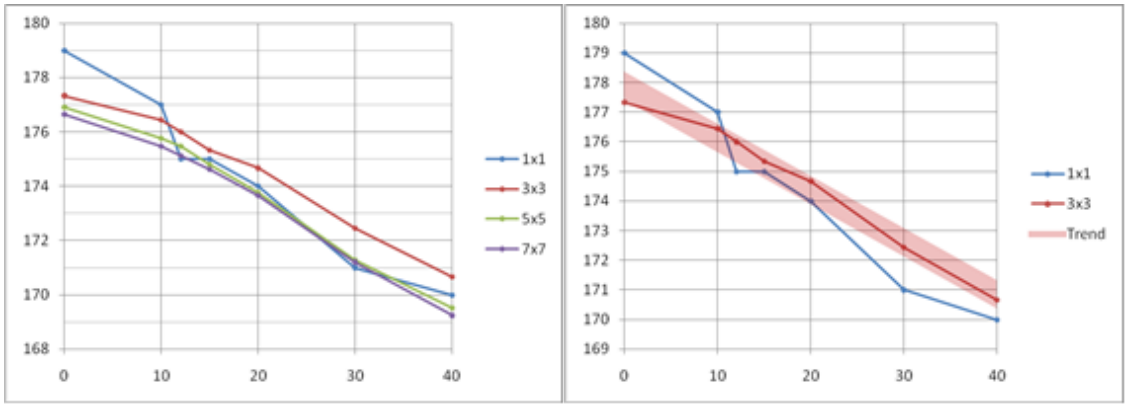

Figure 53: The relative distance against the recorded mean (normalised) disparity values

Figure 53, above, shows the mean pixel values at the different distances, with each line representing a different sized averaging mask. The 1x1 mask only checks one pixel and is thus not smooth like the other values. Most noticeably, it jumps from 177 to 175 between the distances of +10 and +12 mm. The second graph shows a linear line fitted to the 3x3 values. I set the line to be 1 disparity unit thick as a reference. This graph shows that the amount of error with the 3x3 values is within 1 disparity unit; however, the individual pixels (as shown by the 1x1 values) have an error of between 1 and 2 disparity units. The disparity units at this range equate to approximately 6mm of distance (from the graph), so this suggests that the per-pixel depth precision of the stereo correlation is likely between 6 and 12mm at 1.45m from the camera, though the data are insufficient for proper statistical interpretation.

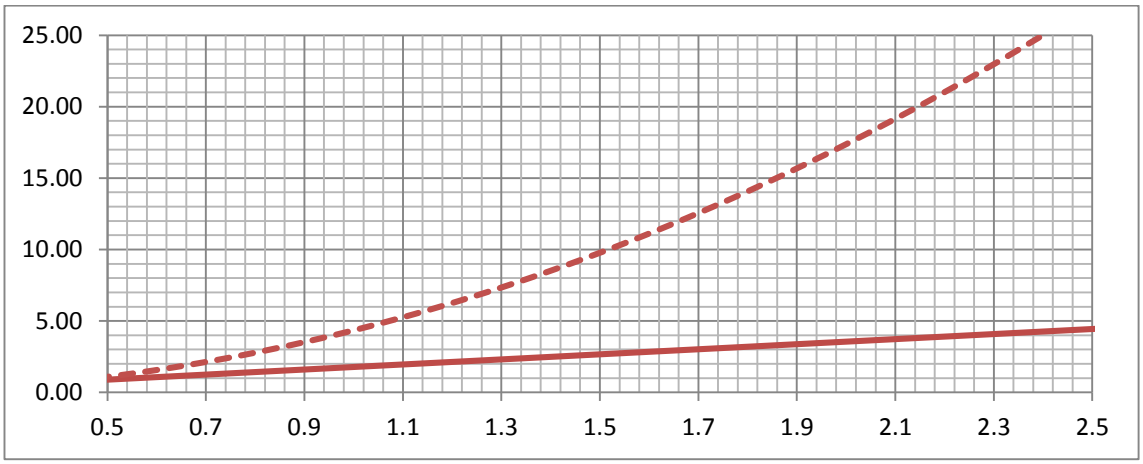

Figure 54: Stated XY (solid) and depth (dashed) precision (mm) of the 65° FoV Bumblebee2 at different depths (m)

However, comparison with the data from Point Grey (graphed in Figure 54), wherein the precision at 1.44 metres is estimated to be 9mm suggests that the results I obtained were no worse than should have been expected.

A.5 Conclusion

In conclusion, I found no major discrepancies between the results I have produced and those advertised by Point Grey. However, it should be understood that the comparisons I made should not be considered perfect due to the differences between the objects and environments being captured in the test images.

Importantly, I was able to show that there were no issues with the Bumblebee's calibration that might have negatively affected the Triclops software's ability to match points. While the results also suggested that it would be possible to obtain the outline of the user's hand, this would still have required unspecified morphological processing. The results also showed that there were no precision issues with the depth values produced by the stereo process.

Appendix B - Platform Evaluation Report

Once I had settled on the design of my first user experiment and which hardware I would use to interpret user actions, it became necessary to consider the best avenues for software implementation. The first and most fundamental question was which development platform to use; i.e., which programming languages and IDEs would provide the easiest path to a system with which the user experiment could be tested. This section documents the options that were available at the time and the rationale behind the choice I made.

B.1 Platforms

Since the more time would be required to implement the system the less time there would be conducting the research proper, the main consideration in the choice of platform is how long it would take to develop code with. This is a function of how easy it would be to write and debug code in that language and how easy it would be to import existing algorithm implementations.

My initial testing and development has consisted of just C++ code, using the freely-available OpenCV image processing library developed by Intel. It uses Intel's Integrated Performance Primitives (IPP) library to achieve high performance on a number of image processing functions. Although this approach leads to fast code, the disadvantage of this approach is that development takes longer due to the time taken to write and debug C++ code.

Another option is BioMOBIUS, which uses the EyesWeb development environment. EyesWeb provides a way to graphically design programs by connecting different blocks together. EyesWeb is designed for interpreting audio and visual data and BioMOBIUS adds a few extra functions and extensions for specific hardware and allows the creation of GUIs to control the programs. The advantage of this approach is a very fast development time, while the code still runs fast (realtime, depending on the blocks used), because the blocks themselves are written in C++. In fact, many of the image processing blocks in EyesWeb are built using OpenCV (though not using many of its functions). However, it is not possible to write code/scripts using the EyesWeb interface. Any additional algorithms or complex controlling code would have to be written in C++ and compiled as an EyesWeb block. The tradeoff is the extra code required for interfacing; however, a lot of this can be copy-and-pasted from previous blocks. One significant limitation of EyesWeb is that it is only available on Windows.

Another graphical development environment is that provided by Simulink, which is based on MATLAB. Most of the important image processing functions are, at a fundamental level, written in C/C++ and some are optimised to use IPP. The MATLAB scripting language is used in at least a small way within the library functions and would most likely be used at some point in the development process. While it is considerably less efficient than C code, it is ideal for high-level control due to its simple and rapidly-developable nature. It is also possible to call C/C++ functions from MATLAB, using MEX-files. While MATLAB and Simulink are intended for prototyping, it is also possible to compile programs and run then in real-time. However, the Image Acquisition toolbox is required to access video data from digital cameras (AUD $325 on the Mathworks website for an individual licence, I'm not sure how much it would be for the school). As with many of the other limitations, it would be possible to get around this requirement by writing custom C++ code to interface with and get the frames from the camera via Windows.

B.2 Included Functions

As can be seen in the table of functions available to each of the platforms below, OpenCV has the most image processing features, and Simulink has slightly more than EyesWeb. The list of functions is not an exhaustive one, but it covers the main ones I might need to use. It should be noted that any feature available in OpenCV could be used in the other development platforms; however a wrapper would have to be written for each, which would be time-consuming in terms of development.

Function		**EyesWeb**	**Simulink**	**OpenCV**
Median Filter		Yes	Yes	Yes
Morphological Operations		Yes	Yes	Yes
Optical Flow	Lucas Kanade	Yes	Yes	Yes
	Horn Shunck	No	Yes	Yes
Histogram-based Tracking		Yes	No	Yes
Blob Labelling/Analysis		Yes	Yes	(via cvBlob library)

Kalman Filter		Yes	Yes	Yes
Edge Detection	Laplace	Yes	Yes	Yes
	Prewitt	Yes	Yes	Yes
	Roberts	Yes	Yes	Yes
	Canny	No	Yes	Yes
Haar Classifier		No	No	Yes
Hough Transforms		No	Yes	Yes
Contour Polygon Extraction		No	No	Yes

B.3 Case Study: Curvature extraction using EyesWeb

To compare the different platforms, I am using the implementation of finding a blob's curvature as the basis for comparison. EyesWeb's standard blocks allow thresholding and other methods to find a blob, but there is no simple way to find the contour and its curvature. I have completed an EyesWeb block that does this, and its results appear to be correct. The input it accepts is a black-and-white image and its output is both a black-and-white image of the contour and a 1x100 matrix of curvature values that can be easily displayed using one of EyesWeb's graphing blocks. The block accepts parameters for changing the scale at which the angles are calculated and toggling between angle and curvature (relative angle) output. Snapshots of the contour and curvature graph can be taken at any time and are saved as JPEG and CSV files.

The time it took to develop the block it was considerably longer than I expected, due to a number of technical issues (EyesWeb block code is dependent on a number of different libraries and has to be built with a certain version of Visual Studio). Also, a lot of the documentation is either lacking details or completely out of date. The process of initialising datatypes is quite complex and it was difficult to debug the custom block as a DLL being run by EyesWeb.

The test system was developed initially to be tested at a resolution of 320x240 pixels. It included a number of different blocks for processing the data and obtaining the hand

blob via background-subtraction. Altogether, these totalled about 5-8ms per frame. The curvature extraction itself only took around 0.15ms per frame. At a resolution of 640 x 480 pixels, the total time, as expected, was around 4 times larger (around 30ms), while the curvature block took 0.43ms on average. This seems to show that EyesWeb can perform at real-time speeds. Ideally, I would like to work at higher frame-rates (around 48fps), so I would want to keep that figure under 20ms, which seems quite possible if I am more careful about which blocks I use.

Since finishing the first, I have written two more blocks for EyesWeb. Although they perform less complex tasks, they both took considerably less time to implement than the first one, mostly because I had already solved the compiling issues and I am now more familiar with how EyesWeb works. Because of this, I expect the time overhead of writing blocks to wrap any future functions would be quite small (<1 hour).

If I were to write more blocks, it would take considerably less time for each because the compiling issues would no longer exist and I am now more familiar with how EyesWeb works. However, there may still be some parts I don't fully understand, so temporary slowdowns remain a possibility.

B.4 Conclusion

Using a graphical programming environment for development would allow for a much faster development cycle. Although there is an overhead involved with adding extra functions, the ability to test different configurations of higher-level functionality without editing code would be extremely useful. The two main options (EyesWeb and Simulink) have different advantages and disadvantages. The main advantage of EyesWeb is that it is available for free for research purposes, whereas Simulink's plugins to make real-time operation and video input possible need to be purchased at significant cost. While Simulink has a wider range of functions, most of these are not useful for my research, and many of those that are could be created by wrapping OpenCV functions.

Although MATLAB's scripting capabilities would make some development easier, using it might result in a slower program and would run the risk of being stuck with MATLAB, even if it became necessary to use another platform.

The restriction to Windows that using EyesWeb would result in is not an issue, since most 3rd-party libraries (such as OpenCV) are available on Windows and the SDK for the Bumblebee2 stereo camera that I intend to use supports Windows.

Taking these different factors into account, I think it would be best to continue using EyesWeb. I can make sure that any code I write myself will be reasonably encapsulated so that even if I had to change platform, it would require not too great an effort.

Appendix C - User Study I Questionnaire

PARTICIPANT FEEDBACK

QUESTIONNAIRE:

Please provide some details:

Age:		
Gender:	☐ M	☐ F
Can we keep the camera recording of your interaction?		
Can we archive the data you provide indefinitely?		

Providing your name and email address is completely optional. Your name will be used to connect your data with that of future tests (if you wish to take part in them). Your email address will be used to inform you about the results of the experiment and further opportunities to participate in similar tests. Choosing to allow us to keep the recording of your interaction will provide us with useful data to help in making improvements to the system; however, this is also completely optional.

What kinds of previous spatial navigation experience do you have? (check any boxes that apply)

☐ Google Maps	☐ Google Earth	☐ GIS Software
☐ Video Games (Strategy)	☐ Video Games (First Person)	☐ CAD/3D Modelling
☐ Other (please specify):		

Please tick which kinds of interfaces you have used for spatial navigation before:

☐ Mouse	☐ Keyboard	☐ 3D Mouse	☐ Gesture

INTERVIEW

PLEASE RATE EACH INTERACTION TECHNIQUE OUT OF 5 FOR EACH CRITERION:

Method	Mouse	3D Mouse	Gesture
Naturalness			
Strain			
Speed			
Accuracy			

DID YOU ENJOY COMPLETING THE TASKS?

IS THERE ANYTHING YOU ESPECIALLY LIKE OR DISLIKE ABOUT ANY OF THE INTERACTION TECHNIQUES?

DO YOU THINK THERE ARE ANY IMPORTANT DIFFERENCES BETWEEN THE INTERACTION TECHNIQUES?

DO YOU THINK THAT MORE EXPERIENCE WITH ANY OF THE METHODS WOULD IMPROVE THE SPEED OR ACCURACY OF YOUR INTERACTION?

Mouse	
3D	
Gesture	

WHICH INTERACTION TECHNIQUE DID YOU PREFER OVERALL AND WHICH DID YOU LIKE THE LEAST OVERALL?

Prefer:

Least Like:

DO YOU HAVE ANY SUGGESTIONS FOR IMPROVEMENTS TO THE GESTURE INTERACTION TECHNIQUE OR A BETTER GESTURE TECHNIQUE?

DO YOU THINK THAT A TWO-HANDED GESTURE SYSTEM WOULD BE BETTER AND HOW WOULD IT WORK?

Appendix D - User Study II Questionnaire

PARTICIPANT FEEDBACK

QUESTIONNAIRE:

Please provide some optional details:

Gender:	☐ M	☐ F
Can we archive the data you provide indefinitely?		

Providing your name and email address is completely optional. Your name will be used to connect your data with that of future tests (if you wish to take part in them). Your email address will be used to inform you about the results of the experiment and further opportunities to participate in similar tests. Choosing to allow us to keep the recording of your interaction will provide us with useful data to help in making improvements to the system; however, this is also completely optional.

What kinds of previous spatial navigation experience do you have? (check any boxes that apply)

☐ Google Maps	☐ Google Earth	☐ GIS Software
☐ Video Games (Strategy)	☐ Video Games (First Person)	☐ CAD/3D Modelling
☐ Other (please specify):		

Please tick which kinds of interfaces you have used for spatial navigation before:

☐ Mouse	☐ Keyboard	☐ 3D Mouse	☐ Gesture
☐ Other (please specify):			

INTERVIEW

PLEASE RATE EACH INTERACTION TECHNIQUE OUT OF 10 FOR EACH CRITERION:

Method	3D Mouse		Gesture	
Naturalness	/10		/10	
Comfort (Lack of Strain)	/10		/10	
Speed	/10		/10	
Accuracy	/10		/10	
Learnability	/10		/10	
Potential	/10		/10	
Appropriateness (Navigation)	/10		/10	
Appropriateness (Tracing Area)	/10		/10	
Appropriateness (Placing Markers)	/10		/10	
Overall	/10		/10	

DID YOU ENJOY THE TEST?

IS THERE ANYTHING YOU ESPECIALLY LIKE OR DISLIKE ABOUT EITHER OF THE INTERACTION TECHNIQUES?

DO YOU THINK THERE ARE ANY IMPORTANT DIFFERENCES BETWEEN THE INTERACTION TECHNIQUES?

DO YOU THINK THAT MORE EXPERIENCE WITH EITHER OF THE INTERACTION TECHNIQUES WOULD IMPROVE THE SPEED OR ACCURACY OF YOUR INTERACTION?

DO YOU HAVE ANY SUGGESTIONS FOR IMPROVEMENTS TO THE GESTURE INTERACTION TECHNIQUES OR A BETTER GESTURE TECHNIQUE?

DID YOU FIND THE DIRECTIONAL TWO-HANDED GESTURE FEATURE USEFUL?

WHAT WAS YOUR OPINION OF THE 3D DISPLAY? DO YOU FEEL IT WAS IT IMPORTANT TO EITHER OF THE INTERACTION TECHNIQUES?

DO YOU THINK USING A DESKTOP MONITOR WOULD HAVE ANY EFFECT ON THE INTERACTION?

DO YOU HAVE ANY OTHER COMMENTS?